1986

Computational strategies for estimation of variance components

Paul Michael VanRaden
Iowa State University

VanRaden, Paul Michael

COMPUTATIONAL STRATEGIES FOR ESTIMATION OF VARIANCE COMPONENTS

Iowa State University PH.D. 1986

PLEASE NOTE:

In all cases this material has been filmed in the best possible way from the available copy. Problems encountered with this document have been identified here with a check mark __✓__.

1. Glossy photographs or pages _____
2. Colored illustrations, paper or print _____
3. Photographs with dark background _____
4. Illustrations are poor copy _____
5. Pages with black marks, not original copy _____
6. Print shows through as there is text on both sides of page _____
7. Indistinct, broken or small print on several pages ✓
8. Print exceeds margin requirements _____
9. Tightly bound copy with print lost in spine _____
10. Computer printout pages with indistinct print _____
11. Page(s) _____ lacking when material received, and not available from school or author.
12. Page(s) _____ seem to be missing in numbering only as text follows.
13. Two pages numbered _____. Text follows.
14. Curling and wrinkled pages _____
15. Dissertation contains pages with print at a slant, filmed as received _____
16. Other _____

_____

_____

University
Microfilms
International

Computational strategies for estimation of variance components

by

Paul Michael VanRaden

A Dissertation Submitted tc the
Graduate Faculty in Partial Fulfillment of the
Requirements for the Degree of
DOCTOR OF PHILOSOPHY

Department: Animal Science
Major: Animal Breeding

Approved:

Signature was redacted for privacy.

In Charge of Major Work

Signature was redacted for privacy.

For the Major Department

Signature was redacted for privacy.

For the Graduate College

Iowa State University
Ames, Iowa

1986

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

INTRODUCTION

Estimation of variance components is one of the more complicated subjects facing animal breeders. Many strategies are available, each having different properties and different computational costs. Ideally, one would like to use a strategy which gives estimates with highly desirable properties without these estimates costing too much. This manuscript outlines procedures for obtaining low-cost, desirable estimates of variance components and gives results of these procedures as applied to both simulated and real data.

Procedures are presented for obtaining upper and lower bounds on restricted maximum likelihood (REML) estimates of variance components in models without relationships at a fraction of the cost of obtaining exact REML estimates. Examples of these estimates from simulated data are presented for a model with herds and sires, a model with herds, sires, and interaction, a cross-classified random model, a multi-trait model, and a model with selection present in the data. These estimates are compared to estimates from an iterated version of an algorithm of Henderson (1980a) which can be used to obtain approximate REML estimates. Both procedures gave satisfactory results in all models except that the algorithm of Henderson was biased substantially by selection.

An approximate REML procedure for models containing relationships is also presented, along with approximate methods for obtaining standard

errors of estimates of variance components for models with and without relationships. Efficient strategies for computing exact REML estimates when this can be done are also discussed. Finally, procedures for estimating accuracies and prediction error variances of sire solutions when the model contains relationships are developed. Applications of many of these procedures to real data problems are included.

REVIEW OF LITERATURE

Estimation of variance components has been and will continue to be one of the major tasks concerning animal breeders. Heritabilities, repeatabilities, and genetic correlations are functions of variance components and are required in order to intelligently design breeding programs, to set up selection index equations, or to rank animals by more sophisticated strategies such as BLUP.

One would think that accurate estimation of variance components would be no problem in animal breeding situations because very large data sets are often available. Unfortunately, the data are almost always unbalanced, often undesigned, and usually contain selection. This can make it impossible to obtain estimates of variance components with desirable properties no matter how many data are available unless the more sophisticated estimation methods are used.

Methods of Estimation and Their Properties

Until 1953, the only available methods of variance component estimation applied only to balanced data or to unbalanced data in very simple designs such as the one-way classification (Freeman (1979)). These methods were based on the analysis of variance (ANOVA) and relied on partitioning mean squares into an amount attributable to error variance and an amount due to variance of the effect in question. These estimates were fairly easy to compute.

Henderson (1953) extended the ANOVA-based approach to unbalanced

data with his Methods I, II, and III. Method I computes fairly simple sums of squares and equates these to their expectations, which are also easy to compute. Method I is valid only for completely random models where the only fixed effect is the overall mean. Method II adjusts the data for fixed effects by least squares and then uses the procedures of Method I on the adjusted data. Method II cannot be used for all models, for instance, those containing interactions between fixed and random effects (Searle (1968)). Method III is more complex but can be used for a wider variety of models. Various reduction sums of squares are computed and equated to expectations. Estimates may not be unique because different sums of squares can be computed. Some reduction sums of squares may require large computing costs. For all three methods, the only guaranteed property is unbiasedness, and even this property may be lost for data containing selection.

A search for methods with more desirable properties led to development of MIVQUE (minimum variance quadratic unbiased estimation) and MINQUE (minimum norm quadratic unbiased estimation). These estimators incorporate prior information about the variance components and are the "best" estimators possible if the priors happen to equal the true parameter values. "Best" in the case of MIVQUE means minimum variance and for MINQUE means minimum norm, but the estimators are identical computationally and differ only on whether normality has been assumed.

MIVQUE theory is primarily attributable to LaMotte (1970, 1971, 1973) while MINQUE was developed by Rao (1971a, 1971b, 1972). Computation of these estimates is rather difficult. Even when the LaMotte quadratic forms and their expectations can be rewritten in terms of the Henderson mixed model equations (Henderson (1986)), repeated inversion of large matrices is still required. Another problem of these estimators is that they are not unique, in the sense that the use of different priors results in different estimates from the same method. A desirable property of MIVQUE is that it eliminates bias due to selection if selection is of the L'y type and if priors equal the true parameters (Henderson (1980b)).

Maximum likelihood (ML) and restricted maximum likelihood (REML) are the final two methods to be discussed. ML is a method widely used in statistics, while REML is similar except that it accounts for degrees of freedom used for estimating fixed effects (Patterson and Thompson (1971)). Neither ML or REML is unbiased except for special cases, but the methods have other desirable properties such as always producing estimates within the parameter space and producing unique estimates which do not depend on the priors used. They are unaffected by selection or assortative mating in the data (Gianola and Fernando (1986)). Animal breeders seem to prefer REML over ML, probably because of the substantial degrees of freedom for fixed effects in their models and the substantial bias this can cause in ML estimation. Harville (1977) has reviewed the development and many of the problems associated with ML and REML estimation.

Computational Aspects and Approximate Strategies

Methods of estimation which provide the most desirable estimates of variance components also usually cost the most. Because these costs are not trivial with large data sets, researchers are often forced to choose between precision and economy. Three options are available for reducing costs: choose a simpler model or estimation method; use a more efficient algorithm to obtain the desired estimate; or use an approximate strategy which mimics the desired method. Many papers in recent years have concerned themselves with these problems by developing new algorithms or by comparing available methods theoretically and by comparing estimates from simulated data with known parameters.

Models can be constructed with varying degrees of complexity depending on how much of the true situation one wants to explain. Often it is necessary to use a simpler model than one would like in order to make estimation of variance components feasible. One example is to use a completely random model instead of a mixed model so that Method I can be used. In a herds and sires model, for instance, herds might be treated as random instead of fixed, in which case the Method I estimate of sire variance would be biased under the assumption that herds are actually fixed (Henderson (1953)), but perhaps this bias would be very small. Another example is to exclude relationships from the model even though animals are related.

Often the same final estimates of variance components can be

obtained by several different computational strategies. Development of efficient algorithms remains a crucial problem for animal breeders even as computing costs decline, simply because data sets grow larger and models grow more complex. Harville (1977) stated that "there is no real hope for finding a single ... algorithm for the ML or REML estimation of θ that will be best, or perhaps even satisfactory, for every application." Nevertheless, there are many strategies and tricks one should know in order to avoid using the worst algorithm.

Very efficient expressions exist for obtaining ML or REML estimates from certain simple models if the data happen to be balanced. For unbalanced data, iterative strategies are required for ML, REML, or approximate REML. One algorithm for REML is to iterate on MIVQUE equations until convergence, provided that convergence occurs within the parameter space (Harville (1977)). A similar algorithm is to use the same quadratic forms as for MIVQUE, but the expectations of these quadratic forms are computed as if the prior was the true value (Schaeffer (1983)). These expectations are simpler to compute than those of MIVQUE, but convergence is slower. This algorithm does have the advantage of keeping estimates within the parameter space.

Iterative algorithms produce a series of estimates. One way to speed convergence is to use boosting, which works by spotting trends in the series of estimates and using these trends to project the final estimate. A simple method is the secant method (Forsythe et al.

(1977)), also called common intercept approach by Schaeffer (1983). Another common method is the relaxation factor (Schaeffer (1983)), which is useful when successive estimates always proceed in one direction. Several other techniques discussed by Harville (1977) are useful when more than one variance component are to be estimated.

A useful trick for working with multi-trait models is to transform the multi-trait system to a set of independent single-trait problems by finding an appropriate rotation. This strategy can be used only if all traits are measured on all animals (Foulley et al. (1982)). Meyer (1985) discussed the use of this transformation as applied to variance component estimation. She presented an algorithm for the no-relationship model which gives rapid convergence but does not guarantee the estimated variance-covariance matrix to be positive definite. Schaeffer (1986) developed an algorithm which does guarantee positive definite estimates but which has very slow convergence.

Schaeffer (1986) discussed another form of rotation which can be used in many situations where all traits are not measured on all animals. This rotation does not reduce the multi-trait system to independent single-trait systems but causes large portions of the matrix to be null, which reduces computing costs. Positive definite estimates can also be guaranteed by this algorithm.

Another transformation for which the effects on variance component estimation have been discussed is tridiagonalization of the

sire coefficient matrix (Taylor et al. (1983)). This transformation is expensive initially but it allows second and later iterations to proceed extremely rapidly. Taylor et al. (1983) achieved rapid convergence using tridiagonalization in conjunction with the EM algorithm of Henderson (equation [18] in Methods Section) and the rotation of traits of Foulley et al. (1982).

Several approximate methods have been developed to obtain reasonable estimates of variance components for less cost. Two of these are MIVQUE (0) and Henderson's simple method (HSM). Both of these mimic the computations of MIVQUE but use simpler quadratics and expectations. MIVQUE (0) is a particular form of MIVQUE where prior values for variances of random effects are all taken to be 0 (Henderson (1980a)). HSM uses quadratics very similar to those of MIVQUE but ignores off-diagonals in these quadratics so that expectations are simpler to obtain (Henderson (1980a)). Both MIVQUE (0) and HSM give unbiased estimates of variance components in the no-selection case.

Harville (1977) suggested some general strategies for approximating REML. A specific approach would be to use an iterated version of HSM, which will be called IHSM. Although little or no theoretical work or simulation studies have been done on IHSM, the method apparently gave reasonable estimates of variance and covariance components in large sets of beef field data (Skaar (1985)) and dairy field data (Hudson and VanVleck (1982)). A problem with approximations like these is that the user is unsure of how close the approximate

estimates are to exact REML estimates. Another problem is that standard errors of the estimates are either unavailable or difficult to compute.

A common strategy for reducing costs is to replace the REML error quadratics by quadratics easier to compute. Two quadratics often used are the within smallest subclass sum of squares (WSS) (Henderson (1980a)) and the residual sum of squares after fitting fixed effects (Skaar (1985)). These quadratics have the advantage of not having to be recomputed at each round of iteration. If data are plentiful, either quadratic may lead to estimates nearly identical to those from the REML error quadratic.

Comparisons of Estimators

Many researchers have reported results from theoretical, actual, or Monte Carlo (simulated) comparisons of estimators. The idea is that the more one knows about the properties and costs of various estimators, the better position one is in to choose a reasonable strategy for any particular problem. A good, brief discussion of the costs of various methods is in Henderson (1980a). Also in this paper, Henderson compared the sampling variances of four different estimators (Method III, MIVQUE, MIVQUE (0), and HSM) for a simulated data set. Conclusions were that HSM performed almost as well as MIVQUE and better than Method III for that data set, and that MIVQUE (0) was an inferior strategy over the range of true variances tested.

Dempfle et al. (1983) compared three of these same methods,

excluding MIVQUE (0), on a larger, real data set. Their conclusions were about the same. MIVQUE had sufficiently smaller variance than Method III to recommend its use where computationally feasible. Where MIVQUE is not computationally feasible, HSM would be a useful second choice, ranking above Method III.

Lin and McAllister (1984) compared Method III, MIVQUE, ML, and REML in simulated data. Three of the methods (Method III, MIVQUE, and REML) produced similar estimates of error variance but ML showed substantial bias. For estimating sire variance, three methods (MIVQUE, ML, and REML) had similar mean squared errors but Method III had larger mean squared error. Both Dempfle et al. (1983) and Lin and McAllister (1984) investigated effects of selection on parameter estimates, but in neither case did they use the L'y type of selection which has been discussed in theoretical papers, for instance Henderson (1980b).

METHODS

All mixed linear models can be represented by the general form

$$y = Xb + Zu + e,$$

with

y - vector of observations,

X - coefficient matrix of fixed effects,

b - vector of unknown fixed effects,

Z - coefficient matrix of random effects,

u - vector of unknown random effects,

e - random residual.

For specific models, b and u may be partitioned to contain any number of individual fixed or random effects. Furthermore, many different assumptions could be used regarding the variances of u and e. Regardless of these assumptions, it is common to absorb the equations for fixed effects into the equations for random effects to yield the following system for estimating $\hat{u}$:

$$(Z'MZ + K)\hat{u} = Z'My . \qquad [1]$$

Here M is introduced as $R^{-1}-R^{-1}X(X'R^{-1}X)^{-}X'R^{-1}$, with the product R times σ_0^2 being the variance of e, and K is defined to be σ_0^2 times the inverse of the variance of u. R and K are usually of simple form and easy to invert.

Bounds on Diagonals of Inverse

Diagonal elements of $(Z'MZ + K)^{-1}$ are required to obtain accuracies and prediction error variances of $\hat{u}$, as well as to obtain REML estimates of variance components. VanRaden and Freeman (1985) presented strategies to obtain bounds on these diagonal elements from simple functions of elements of the original matrix. Strategies were presented only for models in which K can be represented as an identity matrix times a constant, or Ik. Natural extensions of these procedures exist for many models, a few of which will be described briefly here.

Derivation of bounds requires that certain rows and columns of the coefficient matrix be partitioned from the rest of the matrix. This need not be done physically but only conceptually. Let the partitioned matrix be represented

$$Z'MZ + K = \begin{bmatrix} F_{11} + K_{11} & F'_{21} \\ F_{21} & F_{22} + K_{22} \end{bmatrix} \qquad [2]$$

and its similarly partitioned inverse be

$$(Z'MZ + K)^{-1} = \begin{bmatrix} Q_{11} & Q'_{21} \\ Q_{21} & Q_{22} \end{bmatrix} \qquad [3]$$

Partitioning is done such that Q_{11} is of much smaller dimension

than Q_{22}. Bounds for Q_{11} are then found by constructing exact relationships for Q_{11} and substituting larger or smaller matrices into these expressions. A matrix A is defined to be larger than a matrix B if $x'Ax \geq x'Bx$ for all x. Exact expressions for Q_{11} (Hohn, 1973, p. 78) are

$$Q_{11} = [F_{11} + K_{11} - F'_{21}(F_{22} + K_{22})^{-1}F_{21}]^{-1}, \text{ and} \qquad [4]$$

$$Q_{11} = (F_{11} + K_{11})^{-1} + (F_{11} + K_{11})^{-1}F'_{21}Q_{22}F_{21}(F_{11} + K_{11})^{-1}. \qquad [5]$$

The easiest way to obtain bounds on Q_{11} is to realize that both $(F_{22} + K_{22})^{-1}$ and Q_{22} are bounded between the null matrix ϕ and K_{22}^{-1}, which is usually simple to obtain. Substituting ϕ for $(F_{22} + K_{22})^{-1}$ in [4] or for Q_{22} in [5] gives a lower bound

$$Q_{11} \geq (F_{11} + K_{11})^{-1}. \qquad [6]$$

Substituting K_{22}^{-1} into [4] and [5] gives upper bounds [7] and [8], respectively.

$$Q_{11} \leq [F_{11} + K_{11} - F'_{21}K_{22}^{-1}F_{21}]^{-1}, \qquad [7]$$

and

$$Q_{11} \leq (F_{11} + K_{11})^{-1} + (F_{11} + K_{11})^{-1}F'_{21}K_{22}^{-1}F_{21}(F_{11} + K_{11})^{-1}. \qquad [8]$$

VanRaden and Freeman (1985) proved, for the case in which Q_{11} is

a scalar, that the second of the two upper bounds [8] is always the preferred formula. They also presented a third bound strategy for the scalar case which will be generalized here for cases in which Q_{11} is not scalar.

The third bound strategy uses an altered form of the coefficient matrix in which no variance ratio is added to the diagonals of interest, as if the corresponding elements of u were fixed instead of random. This matrix has the form

$$\begin{bmatrix} F_{11} & F_{21}' \\ F_{21} & F_{22} + K_{22} \end{bmatrix} . \qquad [9]$$

Its inverse is denoted

$$\begin{bmatrix} Q_{11}^* & Q_{21}^{*\prime} \\ Q_{21}^* & Q_{22}^* \end{bmatrix} \qquad [10]$$

Assuming that F_{11} is invertible (this may not always be the case), an exact expression can be produced for Q_{11}^* similar to that for Q_{11} in [4].

$$Q_{11}^* = [F_{11} - F_{21}'(F_{22} + K_{22})^{-1}F_{21}]^{-1}. \qquad [11]$$

A direct relationship between Q_{11}^* and Q_{11} is

$$[Q_{11}^{*-1} + K_{11}]^{-1} = [F_{11} - F_{21}'(F_{22} + K_{22})^{-1}F_{21} + K_{11}]^{-1},$$

or

$$[Q_{11}^{*-1} + K_{11}]^{-1} = Q_{11}. \qquad [12]$$

Next, bounds are produced for Q_{11}^* with an expression analogous to [5] and then these bounds on Q_{11}^* are translated to bounds on Q_{11} by using [12]. Of primary interest are the upper and lower bounds on Q_{11}^* when Q_{22}^{*-1} is replaced with K_{22}^{-1} and the null matrix, respectively. From partitioned matrix identities,

$$Q_{11}^* = F_{11}^{-1} + F_{11}^{-1}F_{21}'Q_{22}^* F_{21}F_{11}^{-1}.$$

Because Q_{22}^* is bounded between ϕ and K_{22}^{-1},

$$Q_{11}^* \leq F_{11}^{-1} + F_{11}^{-1}F_{21}'K_{22}^{-1}F_{21}F_{11}^{-1}, \qquad [13]$$

and

$$Q_{11}^* \geq F_{11}^{-1}. \qquad [14]$$

It may not be obvious that a bound on Q_{11}^* will translate directly to a bound on Q_{11} but this is not difficult to show. Suppose L is a matrix larger than Q_{11}^*. The inverse of L must then be smaller than Q_{11}^{*-1}. The expression $(L^{-1} + K_{11})$ must be smaller than $(Q_{11}^{*-1} + K_{11})$ and consequently $(L^{-1} + K_{11})^{-1}$ is larger than $(Q_{11}^{*-1} + K_{11})^{-1}$, which

is Q_{11}, and this completes the proof. Similarly, a lower bound on Q^*_{11} translates to a lower bound on Q_{11}. Bounds on Q_{11} produced by applying relationship [12] to bounds on Q^*_{11} in [13] and [14] are in [15] and [16]. Expression [16] is identical to lower bound [6] derived earlier.

$$Q_{11} \leq [(F_{11}^{-1} + F_{11}^{-1}F'_{21}K_{22}^{-1}F_{21}F_{11}^{-1})^{-1} + K_{11}]^{-1}, \text{ and} \qquad [15]$$

$$Q_{11} \geq [F_{11} + K_{11}]^{-1}. \qquad [16]$$

For an example of how these bound formulas are used, consider a model containing herds, sires, and herd-by-sire interaction, with herds fixed and the other two terms random. Ratios of error to sire variance and error to interaction variance are k_1 and k_2, respectively, and variances of the two random effects and the error term are of the form $I\sigma_i^2$. Mixed model equations are

$$\begin{bmatrix} X'X & X'Z_1 & X'Z_2 \\ Z_1'X & Z_1'Z_1 + Ik_1 & Z_1'Z_2 \\ Z_2'X & Z_2'Z_1 & Z_2'Z_2 + Ik_2 \end{bmatrix} \begin{bmatrix} b \\ \hat{u}_1 \\ \hat{u}_2 \end{bmatrix} = \begin{bmatrix} X'y \\ Z_1'y \\ Z_2'y \end{bmatrix}, \qquad [17]$$

with subscript 1 denoting sires and subscript 2 denoting sire-by-herd interaction.

Bounds on diagonals of the inverse corresponding to sires are most easily obtained by absorbing interaction at the same time as

herds, and then by using procedures developed by VanRaden and Freeman (1985) for a model with just herds and sires. In terms of the current development, this would mean letting F_{11} be simply the diagonal element of $Z_1'M_2Z_1$ for a particular sire and letting F_{21}' represent the remaining elements in that row. M_2 is the absorption matrix including both herds and interaction. Q_{11} is then a scalar representing a diagonal of the inverse and bounds on it should be straightforward from the inequalities presented earlier.

To obtain bounds on diagonals of the inverse corresponding to interaction effects, slightly different procedures must be used. It would be nice to absorb the sire equations into the interaction equations, but this would be expensive to do if the number of sires was large. If only herds were absorbed, leaving sires and interaction in the matrix, one could still apply the bound procedures by isolating a particular row and column at a time. Unfortunately, bounds would be unnecessarily wide because large off-diagonal elements exist in this matrix. Specifically, there is always one element in each row of $Z_2'MZ_1$ as large as the diagonal of $Z_2'MZ_2$.

For tighter bounds from this sire and interaction matrix, rows and columns should be isolated two at a time. A row and column for an interaction effect plus the row and column of the sire involved in the interaction would be partitioned out together. The general strategy is to capture the large off-diagonal elements and their corresponding diagonals in F_{11} so that they can be directly inverted,

with F_{21} containing the more inconsequential off-diagonal elements. Intuitively, tighter bounds are produced because a larger amount of information is in the inverted portion of the matrix.

REML Estimation of Variance Components

For the next three sections, attention will be restricted to models in which each random effect has variance $I\sigma_i^2$. Estimation of σ_0^2 is covered in a later section entitled "Estimators of error variance and their standard errors." Two commonly used formulas exist which can be iterated upon to obtain REML estimates of σ_i^2 in such models (Harville (1977)). These are

$$\hat{\sigma}_i^2 = (\hat{u}_i'\hat{u}_i + tr(C_{ii})\hat{\sigma}_0^2)/n_i, \qquad [18]$$

and

$$\hat{\sigma}_i^2 = \hat{u}_i'\hat{u}_i/(n_i - tr(C_{ii})k_i), \qquad [19]$$

where C_{ii} is a particular portion of $(Z'MZ + K)^{-1}$, n_i is number of levels of effect i, and k_i is $\hat{\sigma}_0^2/\hat{\sigma}_i^2$ from the previous round of iteration. Harville stated that [19] was a "possibly interesting" modification of [18] but offered no help in deciding which one to choose. [18] is derived simply by setting $\hat{u}_i'\hat{u}_i$ equal to its expectation and solving this equation for $\hat{\sigma}_i^2$. [19] is derived by forming the ratio of $\hat{u}_i'\hat{u}_i$ to its expectation and multiplying this ratio by the previous estimate of σ_i^2. Convergence can be attained only when this ratio is 1, or in other words when a value of $\hat{\sigma}_i^2$ has been found such that $\hat{u}_i'\hat{u}_i$ is equal to $n_i\hat{\sigma}_i^2 - tr(C_{ii})\hat{\sigma}_0^2$. Thus, both formulas converge to the same final estimates.

It is useful to know which of these two formulas will converge faster. Suppose the i^{th} quadratic $\hat{u}_i'\hat{u}_i$ is larger than it was expected to be based on the prior. This means that

$$\hat{u}_i'\hat{u}_i > n_i\sigma_i^2 - \text{tr } C_{ii}\sigma_0^2 .$$

Here the two variances represent prior values and not true values. Multiplying both sides by tr $C_{ii}k_i$ gives

$$\text{tr } C_{ii}k_i \; \hat{u}_i'\hat{u}_i > \text{tr } C_{ii}k_i\sigma_i^2(n_i - \text{tr } C_{ii}k_i) ,$$

which further simplifies to

$$\text{tr } C_{ii}k_i \; \hat{u}_i'\hat{u}_i > \text{tr } C_{ii}\sigma_0^2(n_i - \text{tr } C_{ii}k_i) .$$

Adding $(n_i - \text{tr } C_{ii}k_i)\hat{u}_i'\hat{u}_i$ to both sides gives

$$n_i\hat{u}_i'\hat{u}_i > (\hat{u}_i'\hat{u}_i + \text{tr } C_{ii}\sigma_0^2)(n_i - \text{tr } C_{ii}k_i) .$$

Finally,

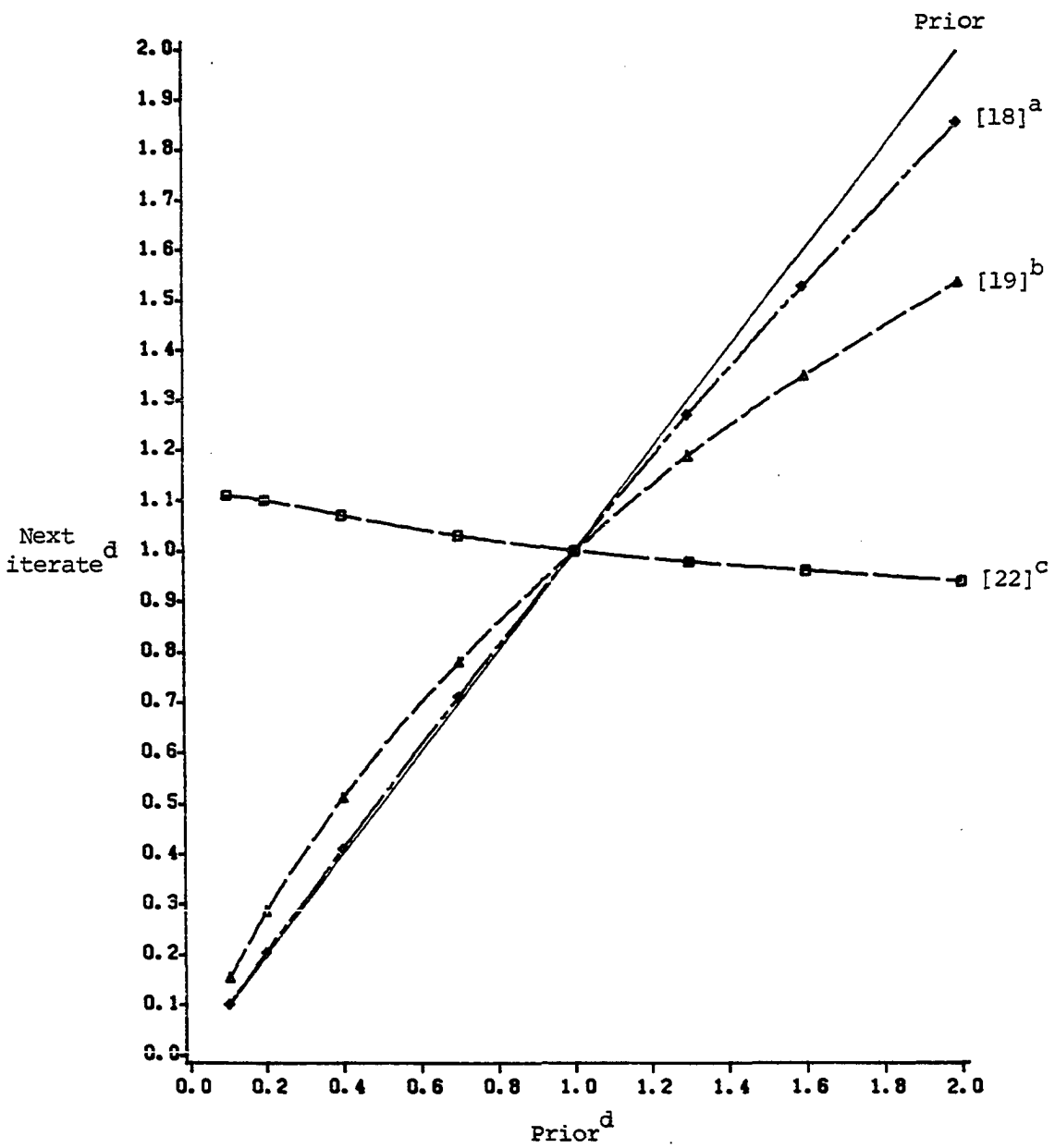
$$\hat{u}_i'\hat{u}_i/(n_i - \text{tr } C_{ii}k_i) > (\hat{u}_i'\hat{u}_i + \text{tr } C_{ii}\sigma_0^2)/n_i .$$

This last inequality demonstrates that, for all prior values of σ_i^2 smaller than the converged value, [19] always produces subsequent estimates higher than those produced by [18]. If directions of

inequalities in the proof are changed, it shows that for prior values of σ_i^2 larger than the converged value, [19] always produces estimates lower than those of [18]. If the inequalities are replaced by equal signs, the same proof shows that the two strategies intersect at the point where the quadratic equals its expectation (based on the prior), and this point is the converged value for both strategies. Behaviors of [18] and [19] when applied to an example data set (from Djemali (1985)) are plotted in Figure 1, along with the behavior of an additional algorithm to be discussed later.

It was shown that [19] always produces an estimate farther away from the prior than [18]. This ensures that [19] will converge faster than [18], provided that [19] is not "overcorrecting" or producing estimates on the opposite side of the converged value from the prior. Figure 1 demonstrates that this is not the case for the example data set at least, but formal proof that [18] and [19] never overcorrect could be obtained by showing that for both strategies the estimate is an increasing function of the prior across all potential values of the prior. This will now be addressed.

Because priors for mixed model equations are usually parameterized as k_i rather than σ_0^2 and σ_i^2, derivatives will be taken first with respect to k_i and then converted to derivatives with respect to σ_i^2 by using the chain rule. Before examining derivatives, it is helpful to diagonalize the mixed model equations by finding eigenvectors and eigenvalues of Z'MZ. This procedure will be discussed in more detail in a later section entitled "exact traces using diagonalization of



$^{a}\hat{\sigma}^2_1 = (\hat{u}_1'\hat{u}_1 + trC_{11}\sigma_0^2)/n_1$. $^{b}\hat{\sigma}^2_1 = \hat{u}_1'\hat{u}_1/(n_1 - trC_{11}k_1)$.

$^{c}\hat{\sigma}^2_1 = \sigma_1^2(\hat{u}_1'\hat{u}_1 - c_1)/(n_1\sigma_1^2 - trC_{11}\sigma_0^2 - c_1)$.

[d]Prior and next iterate for σ_1^2 are both expressed as fractions of the converged value.

Figure 1. Next iterates produced by three REML algorithms

coefficient matrix." The major changes are that u is replaced by u*, Z'MZ is converted to Δ, which is a diagonal matrix whose diagonals are the eigenvalues of Z'MZ, and the new right-hand sides are denoted by r_i.

Diagonalization of Z'MZ is most useful when the model contains only one random effect, such that $K = Ik_1$. The remainder of this section will deal only with such models. Let S contain the normalized eigenvectors of Z'MZ. Mathematically, the changes are

$$u^* = S'u,$$

$$\Delta = S'Z'MZS,$$

and

$$r = S'Z'My.$$

With this notation, [18] can be rewritten as

$$\hat{\sigma}^2_1 = (\hat{u}^{*\prime}\hat{u}^* + tr(\Delta + Ik_1)^{-1}\hat{\sigma}^2_0)/n_1. \qquad [18b]$$

Similarly, [19] can be rewritten as

$$\hat{\sigma}^2_1 = \hat{u}^{*\prime}\hat{u}^*/(n_1 - tr(\Delta + Ik_1)^{-1}k_1). \qquad [19b]$$

In a later section entitled "Exact traces using diagonalization of coefficient matrix" it will be shown that [18b] and [19b] produce iterates and final solutions identical to [18] and [19], respectively, in the more general case of multi-trait models containing single random effects.

The derivative of [18b] with respect to k_1 is

$$\frac{\partial[18b]}{\partial k_1} = \frac{\partial[\hat{u}^{*\prime}\hat{u}^{*} + \mathrm{tr}(\Delta + Ik_1)^{-1}\hat{\sigma}_0^2]/n_1}{\partial k_1} .$$

Although $\hat{\sigma}_0^2$ may be to some degree a function of k_1, we will temporarily assume it to be a known constant. Derivatives needed to evaluate the above expression are

$$\frac{\partial(\hat{u}^{*\prime}\hat{u}^{*})}{\partial k_1} = \frac{\partial[r'(\Delta + Ik_1)^{-2}r]}{\partial k_1}$$

$$= -\,2r'(\Delta + Ik_1)^{-3}r,$$

and

$$\frac{\partial[\mathrm{tr}(\Delta + Ik_1)^{-1}\hat{\sigma}_0^2]}{\partial k_1} = -\,\mathrm{tr}(\Delta + Ik_1)^{-2}\hat{\sigma}_0^2.$$

Both derivatives are obviously negative for all valid (positive) choices of k_1, so $\partial[18b]/\partial k_1$ is negative. But the desired quantity is the derivative of $\hat{\sigma}_1^2$ with respect to σ_1^2 rather than k_1. This is accomplished with use of the chain rule

$$\partial[18b]/\partial\sigma_1^2 = (\partial[18b]/\partial k_1) \cdot (\partial k_1/\partial\sigma_1^2) .$$

The derivative of k_1 with respect to σ_1^2 is

$$\partial k_1/\partial \sigma_1^2 = \partial(\sigma_0^2/\sigma_1^2)/\partial \sigma_1^2,$$

$$= -\sigma_0^2/(\sigma_1^2)^2,$$

which is negative for all choices of σ_1^2. The chain rule shows that $\partial[18b]/\partial \sigma_1^2$ is the product of two negative factors, so it is always positive. This confirms that estimates produced by [18b] are always an increasing function of the prior.

The derivative of [19b] is handled in much the same way, but is a bit more tedious.

$$\frac{\partial[19b]}{\partial k_1} = \frac{\partial[\hat{u}^{*\prime}\hat{u}^*/(n_1 - \mathrm{tr}(\Delta + Ik_1)^{-1}k_1)]}{\partial k_1}.$$

Because both the numerator and denominator are functions of k_1, the familiar expression $\partial(f/g) = (g\partial f - f\partial g)/g^2$ must be used. Because the objective is to determine only the sign of this expression, the term g^2 can be brought to the left side and ignored as this term is always positive. Evaluation of the derivative gives

$$g^2\partial[19b]/\partial k_1 = (n_1 - \mathrm{tr}(\Delta + Ik_1)^{-1}k_1)(-2)r'(\Delta + Ik_1)^{-3}r$$

$$- \hat{u}^{*\prime}\hat{u}^*(-1)\mathrm{tr}[\Delta(\Delta + Ik_1)^{-2}].$$

Further substitution and algebra gives

$$g^2\partial[19b]/\partial k_1 = -2tr[\Delta(\Delta + Ik_1)^{-1}]r'(\Delta + Ik_1)^{-3}r$$

$$+ tr[\Delta(\Delta + Ik_1)^{-2}]r'(\Delta + Ik_1)^{-2}r .$$

This derivative is the sum of one positive term and one negative term, so the overall sign is unclear. For balanced data, however, the sign can be clearly determined. With balanced data, all diagonals of Δ would equal some constant, say w_1, such that $\Delta = Iw_1$. Consequently, the expression above would reduce to

$$g^2\partial[19b]/\partial k_1 = -2n_1w_1/(w_1+k_1) \cdot r'r/(w_1+k_1)^3$$

$$+ n_1w_1/(w_1+k_1)^2 \cdot r'r/(w_1+k_1)^2 ,$$

and finally,

$$g^2\partial[19b]/\partial k = -n_1w_1r'r/(w_1+k_1)^4 .$$

Thus, the derivative of [19b] with respect to k_1 is always negative for balanced data. Experience with unbalanced data sets indicates that this property often holds for unbalanced data sets as well, but the failure of the proof indicates that this may not always be the case. If $\partial[19b]/\partial k_1$ is negative, the chain rule can then be used to show $\partial[19b]/\partial\sigma_1^2$ is positive as was done with [18b]. Properties

claimed for [18b] and [19b] must also hold for [18] and [19] because these formulas are equivalent.

This completes proof that, with balanced data, [19] always converges faster than [18]. For unbalanced data, though the proof was not conclusive, it is reasonable that the same statement holds for most data sets of interest. Thus, [18] should not be used because it offers no advantages to [19] and converges more slowly.

Derivation of New REML Algorithm

In many situations, neither [18] nor [19] provide convergence as fast as one would like. For instance, in balanced data it is known that iterative MIVQUE and certain other strategies will converge in one round, whereas [18] and [19] will not (Harville (1977)). From this theoretical basis, and also from practical experience (Schaeffer (1983)), it can be assumed that iterative MIVQUE converges faster in unbalanced data too. The intent here is to find a REML algorithm which converges with about the same speed as iterative MIVQUE but which retains the lower cost per iteration present in [18] and [19].

Iterative MIVQUE, [18], and [19] all use the same quadratic forms but use different procedures for computing expectations. [18] and [19] assume that prior values are equal to true values, whereas iterative MIVQUE does not. The expectation of $\hat{u}^{*\prime}\hat{u}^{*}$ for MIVQUE, for the case where a single random effect exists in the model, and again using the diagonalized system, is

$$E(\hat{u}^{*\prime}\hat{u}^{*}) = E(r'(\Delta + Ik_1)^{-2}r),$$

$$= tr[(\Delta + Ik_1)^{-2}Var(r)].$$

To evaluate this expression we need Var(r), which is

$$Var(r) = Var(S'Z'My),$$

$$= S'Z'M(I\sigma_0^2 + ZZ'\sigma_1^2)MZS.$$

This can be rewritten in terms of Δ if $I = SS'$ is substituted between Z and Z'. This gives

$$Var(r) = S'Z'MZS\sigma_0^2 + S'Z'MZSS'Z'MZS\sigma_1^2,$$

$$= \Delta\sigma_0^2 + \Delta^2\sigma_1^2.$$

Substituting this into the expectation above gives

$$E(\hat{u}^{*\prime}\hat{u}^{*}) = tr[(\Delta + Ik_1)^{-2}\Delta\sigma_0^2 + (\Delta + Ik_1)^{-2}\Delta^2\sigma_1^2]$$

Once diagonalization has been done, additional computations to evaluate this trace are trivial.

If the system is not diagonalized, and again for models containing only a single random effect ($K = I k_1$), the expectation is

$$E(\hat{u}'\hat{u}) = \mathrm{tr}[Z'MZ(Z'MZ + I k_1)^{-2}\sigma_0^2 + (Z'MZ)^2(Z'MZ + I k_1)^{-2}\sigma_1^2] . \qquad [20]$$

Even in this relatively simple case, computations to evaluate this expression could be expensive. If the model contains more than one random effect, expectations of $\hat{u}_i'\hat{u}_i$ become even more tedious and involve not only σ_0^2 and σ_i^2, but all the other variance components as well (Schaeffer (1983)). Though the MIVQUE expectations are tedious, it is sometimes possible to obtain inexpensive approximations to them. For instance, if all matrices appearing in [20] have large diagonals relative to off-diagonals, the expectation can be approximated by

$$E(\hat{u}'\hat{u}) \approx \sum_j d_j/(d_j + k_1)^2\sigma_0^2 + \sum_j d_j^2/(d_j + k_1)^2\sigma_1^2 , \qquad [21]$$

where d_j is the j^{th} diagonal elementof Z'MZ. This expression looks very similar to that for the diagonalized system and also to the expectation that would be computed for Henderson's simple method (see Henderson (1980a)).

A new REML algorithm can be obtained by first subtracting from $\hat{u}_i'\hat{u}_i$ the portion of its expectation expected to come from error

variance, which will be denoted c_i, and then by deriving an expression in the manner of [19]. This is done by dividing $\hat{u}_i'\hat{u}_i - c_i$ by its expectation, which is simply $E(\hat{u}_i'\hat{u}_i) - c_i$, and then by multiplying this ratio by σ_i^2. This gives

$$\hat{\sigma}_i^2 = \sigma_i^2(\hat{u}_i'\hat{u}_i - c_i)/(n_i\sigma_i^2 - trC_{ii}\sigma_0^2 - c_i), \qquad [22]$$

where σ_i^2 and σ_0^2 again refer to priors, not true values, and c_i can be approximated by $\sum_j d_j/(d_j+k_i)^2\sigma_0^2$, where summation is over the appropriate diagonals. Other approximate formulas could be used to obtain c_i; the choice of these will affect only rate of convergence but not final converged values.

If [22] converges it will converge to the same value as [18] and [19]. This is because, for all three strategies, the converged value is the point at which each $\hat{u}_i'\hat{u}_i$ equals its expectation. For [18] and [19], each iterate remains positive if iteration is started with positive priors, but this is not necessarily true for [22]. The statement could be made to be true for [22], though, if the user makes sure that each c_i is smaller than the corresponding $\hat{u}_i'\hat{u}_i$ and also its corresponding expectation at each iteration. If c_i happened to be larger than $\hat{u}_i'\hat{u}_i$, c_i could be set equal to .9 times $\hat{u}_i'\hat{u}_i$ instead, for instance.

Bounds on REML Estimates

All three iterative algorithms presented for obtaining REML estimates of variance components ([18], [19], and [22]) require computing $\hat{u}_i'\hat{u}_i$ and trC_{ii} for each component at each round of iteration. Costs of computing $\hat{u}_i'\hat{u}_i$ rise in proportion to the square of number of equations if Gauss-Seidel iteration is used, whereas the cost of inverting $(Z'MZ + K)$ to obtain needed traces rises in proportion to the cube of number of equations. This means that, in large systems, the major difficulty is to obtain trC_{ii}. This difficulty can be overcome by computing only bounds on trC_{ii} rather than exact values. For convenience, the average of upper and lower bounds can be used as an approximation for trC_{ii}. This average of bounds might cause biased estimates of variances but at least the magnitude of the bias would be known.

A trace is simply the sum of diagonal elements of a matrix, so a sum of upper bounds on these diagonal elements would constitute an upper bound on the trace, and similarly for a lower bound. Bounds on REML estimates of variance components do not follow quite as easily from bounds on trC_{ii}, however, because of the iterative nature of REML estimation. If only one variance component is to be estimated, though, the procedure is straightforward.

Suppose that one variance component, σ_1^2, is to be estimated, and that σ_0^2 is either assumed known or is held constant across iterations. Algorithms [18], [19], and [22] then produce new values of $\hat{\sigma}_1^2$ as functions only of the σ_1^2 used as a prior, as demonstrated in Figure 1. Converged values are the points at which these curves

intersect the line $\hat{\sigma}_1^2 = \sigma_1^2$ which, as pointed out earlier, all coincide.

Now suppose that an upper bound on trC_{11} is substituted for trC_{11} in [18], [19], or [22]. Inspection of these formulas reveals that, in all cases, higher values of $\hat{\sigma}_1^2$ will be produced than those produced by using the exact trace. (This assumes that denominators of [19] and [22] remain positive, which will be the case if [8] rather than [7] is used as an upper bound formula and if c_i is checked to make sure it is not too large.) Because of these higher values, curves plotted over a range of values for the prior σ_1^2 and which use upper bounds on trC_{11} must intersect the line $\hat{\sigma}_1^2 = \sigma_1^2$ at higher points than curves derived from the exact traces. Thus, iteration until convergence using an upper bound on trC_{11} will result in an upper bound on the REML estimate of σ_1^2 and similarly for a lower bound. Figure 2 plots expression [19] when trC_{11} is replaced by either an upper or lower bound. The exact form of [19] is not plotted but it must lie between the upper and lower curves plotted.

When more than one variance component is to be estimated, the situation grows more difficult. From an initial set of priors k_i, one would obtain upper and lower bounds for each component at the first round of iteration. Then, for the next iterate one is faced with many potential combinations of upper and lower bounds that could be used as priors for the next round. To obtain bounds on eventual REML estimates, all combinations of upper and lower bounds may need to be investigated.

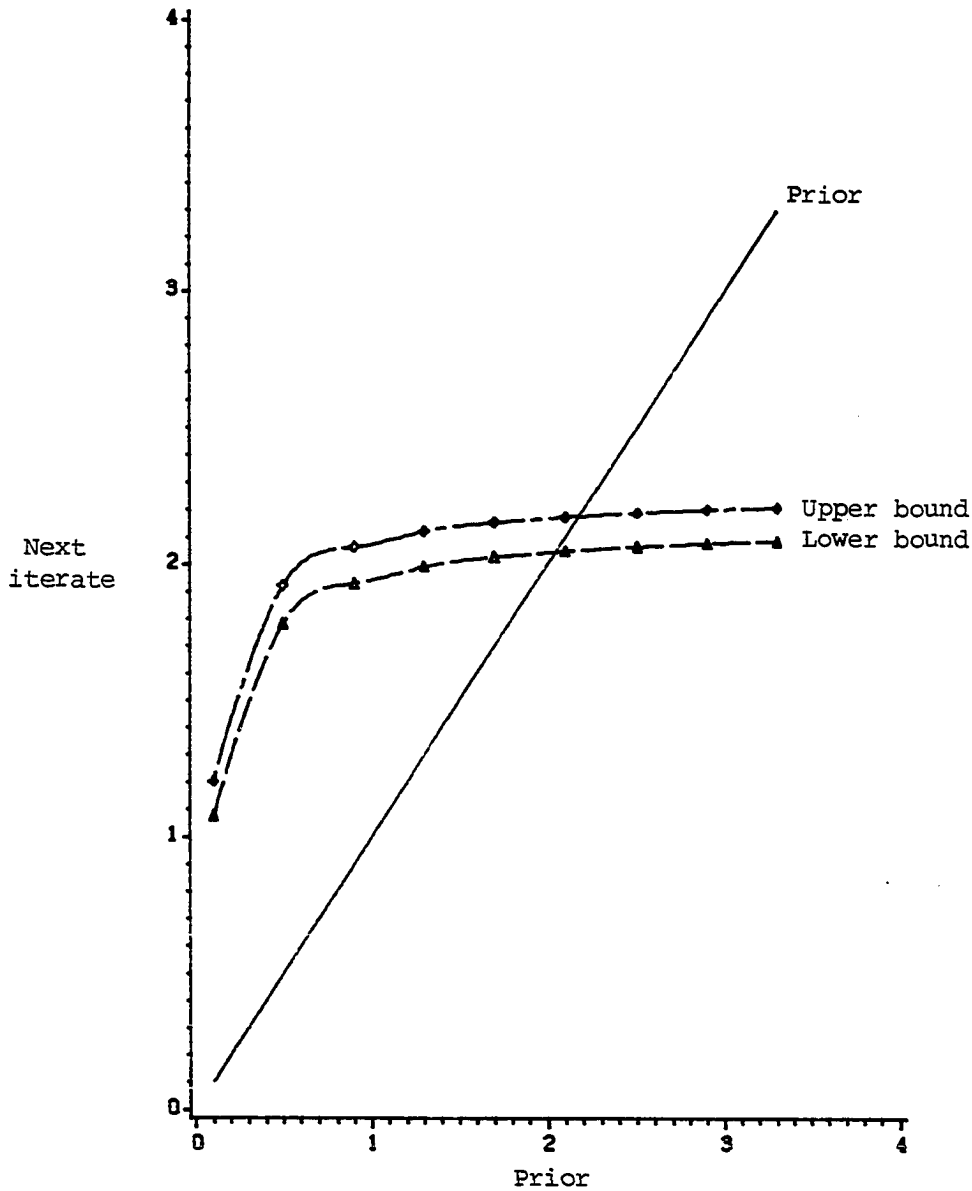


Figure 2. Next iterates produced by upper and lower bounds on formula [19]

The two-variance component model is instructive. One could think of assuming the first component known and holding it constant while iterating until convergence on the second, and vice versa. Two curves could then be plotted. These would be the converged values of $\hat{\sigma}_1^2$ plotted against a range of given values of σ_2^2 and the converged values of $\hat{\sigma}_2^2$ for various given values of σ_1^2. The intersection of these two curves is the REML estimate, provided that this occurs within the parameter space.

Upper and lower bounds can be obtained for either of these two curves from procedures presented earlier. The overlapping region contained between the two sets of upper and lower bounds will contain the exact REML estimate, as demonstrated in Figure 3. But to report such regions in which exact estimates might lie and to have these accepted by users who want only to find estimates could be a problem. A useful procedure would be to report only the minimum and maximum points of the region with respect to σ_1^2 as bounds on $\hat{\sigma}_1^2$ and likewise for $\hat{\sigma}_2^2$. Another useful procedure would be to report only averages of the bounds as an approximation to the exact REML estimate.

An example algorithm for computing minimum and maximum points of a region for a two-variance component model is from a model with cross-classified random effects. At each round of iteration after the first, values of $\hat{u}_1'\hat{u}_1$ and $\hat{u}_2'\hat{u}_2$ are computed using each of the four combinations of upper and lower bounds as priors, giving eight quadratics. These are matched with appropriate upper or lower bounds

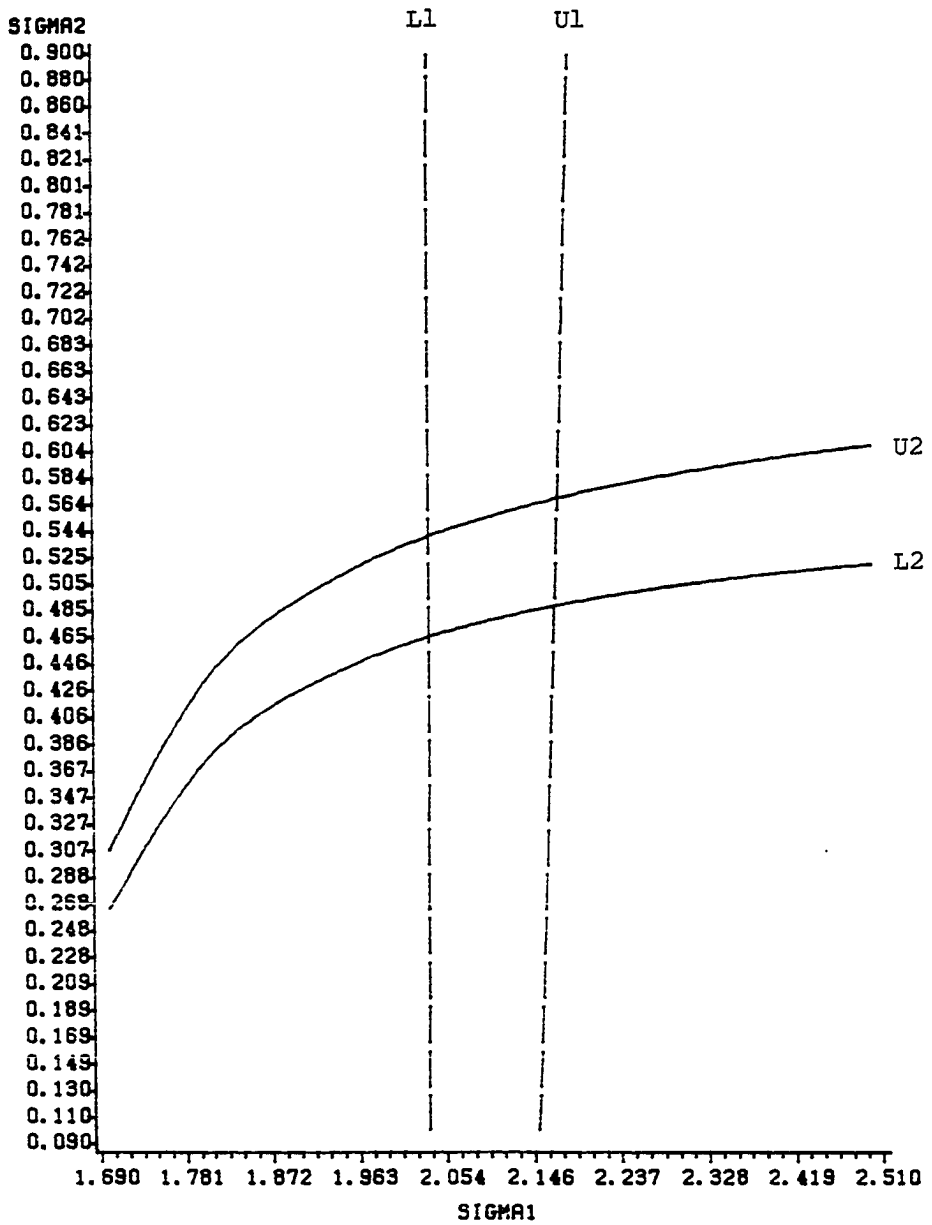


Figure 3. Upper and lower bounds for a cross-classified random model

on expectations computed as if each combination were in turn the true values. The very highest and very lowest estimates produced from these combinations of quadratics and expectations are bounds for the next round. This procedure is obviously complicated and would grow even more so with a larger number of variance components. Thus, computing only averages of bounds may be a more reasonable strategy.

Approximate Bounds for Relationship Models

Models which contain relationships among elements of u are much more difficult to deal with. In the previous three sections, attention was confined to models in which all random effects have variance $I\sigma_i^2$. A more general situation is where at least one random effect has variance $A\sigma_i^2$. In animal breeding problems, elements of A are usually taken to be numerators of Wright's coefficient of relationship among animals (Henderson (1975)). Two serious problems occur when one attempts to compute bound formulas in models where relationships are included.

The first problem is that, for bound formulas such as [7], [8], or [15], the inverse of K_{22} is required. In the no-relationship model K_{22} is simply a scalar times I, but in the relationship model K_{22} contains all the rows and columns of A^{-1} not singled out for inclusion in K_{11}. Thus, K_{22}^{-1} would be similar to, but not the same as, an A matrix for the sires not included in K_{11} and would be difficult to obtain. Furthermore, bound formulas would no longer

involve simple sums of squares of off-diagonal elements but rather more complicated quadratic relationships.

The second problem is that the REML quadratics and expectations become more difficult when relationships are included. For an effect having variance $A\sigma_i^2$, the appropriate quadratic is $\hat{u}_i'A^{-1}\hat{u}_i$, and its expectation is $\sigma_i^2(n_i - tr(A^{-1}C_{ii})k_i)$ (Schaeffer (1983)). Thus, not only the diagonals of $(Z'MZ + K)^{-1}$ but also certain of the off-diagonals are needed, i.e. those corresponding to nonzero off-diagonal elements of A^{-1}. While it is possible to estimate these off-diagonals, easy-to-compute upper and lower bound formulas similar to those for diagonals are not possible. Computing $\hat{u}_i'A^{-1}\hat{u}_i$ is usually not expensive because $\hat{u}_i$ can be obtained again by Gauss-Seidel iteration.

A strategy which avoids both problems is to transform u to u* by choosing a transformation such that Var(u*) is diagonal. Simple bounds for the no-relationship model can then be applied and only diagonal elements of the inverse are required. Drawbacks to this strategy, though, are that pre- and post-multiplying Z'MZ by a transformation matrix can be expensive and inconvenient (because the $\hat{u}^*$ solutions obtained are not those needed for ranking animals) and that this procedure creates some very large off-diagonal elements which cause the bounds to spread apart.

Another approach would be to include in the inverted portion of the matrix (that corresponding to K_{11}) the diagonals and off-diagonals of all animals related to the one in question, or perhaps

just the closest relatives. Next, assume that animals not in the inverted block are unrelated to those in the inverted block and to each other. This latter assumption is not very realistic, but is required so that a diagonal matrix (Ik_i) can be substituted for K_{22}. Then proceed to use bound formulas as before. Estimates of needed diagonals and off-diagonals of $(Z'MZ + K)^{-1}$ will be produced from the bounds and these can be multiplied by appropriate nonzero elements of A^{-1} to yield an estimate of $tr(A^{-1}C_{ii})$ fairly cheaply.

Estimates produced from this last strategy are not true bounds, however, and the procedure is messy to work with if equations are not stored in core. For instance, computing crossproducts of off-diagonals of a son with off-diagonals of his sire would be difficult if equations were not stored in this sequence and if only one equation was to be read in at a time. A way to avoid this would be to use only the lower bound formula, which essentially ignores off-diagonal information.

A particular algorithm was developed to inexpensively estimate $tr(A^{-1}C_{ii})$ even when the equations are too numerous to be stored in core. As coefficients in each row of $Z_i'MZ_i$ and $A^{-1}k_i$ are read, positions of either all nonzero elements of A^{-1} or the N elements with largest absolute values are recorded. Often there will be just three nonzero coefficients which will include diagonal and two off-diagonals corresponding to the sire and maternal grandsire, if these are the relatives used. Individuals with sons and grandsons may have numerous

large off-diagonals. Capturing all of these would probably not be critical, so N could be set to something like 25. Then the equations would need to be read in again so that the remaining elements required to fill out the NxN matrices for each animal could be determined. Each NxN matrix is then inverted and pre-multiplied by the corresponding NxN segment of A^{-1}. The appropriate diagonals are then summed to estimate $tr(A^{-1}C_{ii})$.

Exact Traces Using Diagonalization of Coefficient Matrix

For moderate sized problems, it might be advantageous to initially incur the fairly high cost of diagonalizing Z'MZ so that subsequent iterations could proceed rapidly (Dempster et al. (1984)). This might be especially profitable in multi-trait models not containing relationships, where several inversions of matrices of the form $Z'MZ + Ik_i$ are needed in each round. Consider the common situation in which each animal is measured for t traits and the X and Z incidence matrices are identical for each trait. For the i^{th} trait, then, the model is

$$y_i = Xb_i + Zu_i + e_i.$$

With no loss of generality, this model can be rewritten

$$y_i = Xb_i + ZSS^{-1}u_i + e_i,$$

where S is a square matrix whose columns are the normalized eigenvectors of Z'MZ. Easier notation is achieved by realizing that

$S^{-1} = S'$ (Graybill (1983)). Single-trait mixed model equations after absorption of fixed effects for the i^{th} trait are then

$$(S'Z'MZS + Ik_i)(S'\hat{u}_i) = S'Z'My_i,$$

or

$$(\Delta + Ik_i)(S'\hat{u}_i) = S'Z'My_i,$$

where Δ is a diagonal matrix whose diagonals are the eigenvalues of $Z'MZ$. Let $S'u_i = u_i^*$ and correspondingly $S'\hat{u}_i = \hat{u}_i^*$. The variance of u_i^* is $S'(I\sigma_i^2)S = I\sigma_i^2$ because $S'S = I$. Because $Var(u_i^*) = I\sigma_i^2$, the quadratic form for estimating σ_i^2 in the transformed model would be $\hat{u}_i^{*\prime}\hat{u}_i^* = \hat{u}_i'SS'\hat{u}_i = \hat{u}_i'\hat{u}_i$. Thus, σ_i^2 will be estimated identically in the transformed and original models because equivalent quadratic forms are used.

For models with relationships, a different transformation should be used. Let the transformed model be

$$y_i = Xb_i + ZTT^{-1}u_i + e_i,$$

with $Var(u_i) = A\sigma_i^2$. T must be chosen so that $T'Z'MZT = \Delta^+$ and $T'A^{-1}T = I$, with Δ^+ being a diagonal matrix. Such a T can be found by first choosing a lower triangular matrix L such that $LL' = A$, or identically $L'^{-1}L^{-1} = A^{-1}$. Then find S^+ which has as its columns the normalized eigenvectors of $L'Z'MZL$. The transformation matrix

T is then LS^{+}.

The appropriate quadratic for estimating σ_i^2 in this transformed model is $\hat{u}_i^{+\prime}\hat{u}_i^{+}$, where $\hat{u}_i^{+} = T^{-1}\hat{u}_i$. Substitution of this identity gives $\hat{u}_i'T'^{-1}T^{-1}\hat{u}_i$. But $T'^{-1}T^{-1} = A^{-1}$, so the quadratic is $\hat{u}_i'A^{-1}\hat{u}_i$ which is the same quadratic required for the untransformed model. Again we have proved that σ_i^2 is estimated identically whether or not the transformation is used.

Estimators of Error Variance and Their Standard Errors

This section assumes $\text{Var}(e) = I\sigma_0^2$ rather than the more general $R\sigma_0^2$. REML estimates of error variance in such models are based on the quadratic

$$y'y - y'Z\hat{u} - y'X\hat{b}, \qquad [23a]$$

(Harville (1977)). Computing this quadratic at each round of iteration would require either that the entire data be read in again or that equations for fixed effects be stored so that $\hat{b}$ can be reestimated at each round of iteration. Such computations are generally feasible, but if absorption has been done, [23a] can be rewritten as [23b] which is a more easily computable form. Algebraic rearrangement of mixed model equations gives $\hat{b} = (X'X)^{-}(X'y - X'Z\hat{u})$. Substituting this quantity into [23a] yields

$$y'y - y'Z\hat{u} - y'X(X'X)^{-}(X'y - X'Z\hat{u}), \text{ or}$$

$$y'y - y'Z\hat{u} - y'X(X'X)^{-}X'y + y'X(X'X)^{-}X'Z\hat{u}.$$

This quadratic can be reorganized to a simpler quadratic in terms of $M = I - X(X'X)^{-}X'$, which is

$$y'(I-X(X'X)^{-}X')y - y'(I-X(X'X)^{-}X')Z\hat{u},$$

or

$$y'My - y'MZ\hat{u}. \qquad [23b]$$

Expectations of [23a] and [23b] are both equal to $(N-\text{rank}(X))\sigma_0^2$, where N is the length of y (Schaeffer (1983)). Thus, $\hat{\sigma}_0^2$ at each round is obtained by dividing [23a] or [23b] by N-rank(X).

Another quadratic for estimating error variance is the within smallest subclass sum of squares (WSS), or y'M*y, where $M^* = I - X^*(X^{*\prime}X^*)^{-}X^{*\prime}$ and X* is the incidence matrix for a model with smallest subclasses treated as fixed being the only effect. The expectation of y'M*y is

$$E(y'M^*y) = E(y')M^*E(y) + \text{tr}[M^*\text{Var}(y)],$$

$$= b'X'M^*Xb + \text{tr}[M^*(I + ZK^{-1}Z')\sigma_0^2].$$

Both M*X and M*Z are null, so this simplifies to

$$E(y'M^*y) = \text{tr}[M^*]\sigma_0^2.$$

Because M* is idempotent,

$$E(y'M^*y) = \text{rank}(M^*)\sigma_0^2.$$

Then, $\hat{\sigma}_0^2$ is obtained by

$$\hat{\sigma}_0^2 = y'M^*y/\text{rank}(M^*) .$$

Variances of the REML estimator and the WSS estimator are simply $2\sigma_0^4$ divided by degrees of freedom of the estimator. Derivation of these requires that, for the REML quadratic, exact values of σ_0^2/σ_i^2 are used in computing $\hat{u}$. This would not be the case but is probably a good approximation in large data sets. Standard errors (s.e.) of the REML and WSS estimates of error variance would be, respectively,

$$\text{s.e.}(\hat{\sigma}_0^2) = \sigma_0^2\sqrt{2/(N-\text{rank}(X))}, \qquad \text{(REML)}$$

$$\text{s.e.}(\hat{\sigma}_0^2) = \sigma_0^2\sqrt{2/\text{rank}(M^*)} . \qquad \text{(WSS)}$$

Error Quadratics in Diagonalized Models

Diagonalization of Z'MZ and simultaneous diagonalization of Z'MZ and A^{-1} have no effect on the WSS quadratic. The expectation of WSS does not involve u because M*Z is null. Thus, the expectation of WSS does not involve either u* or u^+ because these enter the expectation as functions of M*ZS and M*ZT, respectively, and both of these must be null if M*Z is null.

The REML error quadratic can be rewritten in terms of the diagonalized models quite easily. Substitution of $I = SS'$ or $I = TT^{-1}$ into [23b] leads to [24a] and [24b], respectively. These error

quadratics are

$$y'My - y'MZSS'\hat{u}, \text{ which equals}$$

$$y'My - y'MZS\hat{u}^*, \text{ and} \qquad [24a]$$

$$y'My - y'MZTT^{-1}\hat{u}, \text{ which equals}$$

$$y'My - y'MZT\hat{u}^+. \qquad [24b]$$

Because both [24a] and [24b] are equivalent to [23b], their expectations and standard errors must equal those of [23b] as well.

Approximate Standard Errors of $\hat{\sigma}_i^2$

Often it is desirable to present standard errors or confidence intervals of point estimates along with the estimates. For variance component estimation, this can be a difficult task. REML estimation with unbalanced data involves iterated solutions to a constrained, nonlinear set of equations, so derivation of exact standard errors is nearly impossible. Asymptotic standard errors can be computed by using certain assumptions about the data, but whether these assumptions are reasonable for real data sets can be questioned (Harville (1977)). The intent here is to find some easily understood method for obtaining approximate standard errors of variance component estimates.

For certain balanced designs and with certain noniterative methods of estimation, standard errors can be computed easily. To take advantage of this, one might first attempt to calculate the size

of a balanced data set which would contain an amount of information equivalent to that in the unbalanced data set at hand. Then, one could compute standard errors as if an unbiased, noniterative method of estimation had been used on this hypothetical, balanced data set. Resulting estimates of standard errors should be reasonable over much of the parameter space but might be upwardly biased near constraints on the parameters, such as the constraint that estimated variances cannot be negative. This is because unbiased procedures which allow negative estimates necessarily have larger standard errors than similar procedures which constrain the estimates to be positive.

To equate data sets based on amount of information, a measure of information is needed. One measure would be the sum of variances of $\hat{u}_i$, or $(n_i - trC_{ii}k_i)\sigma_i^2$, a quantity fundamental to REML estimation as can be seen in [19]. A balanced data set is then envisioned which would have the same number of levels (n_i) and equal information, or

$$(n_i - trC_{ii}k_i)\sigma_i^2 = (n_i - trC^*_{ii}k_i)\sigma_i^2,$$

where C^*_{ii} represents a matrix analogous to C_{ii} but derived from balanced data. This condition is equivalent to requiring that trC_{ii} equal trC^*_{ii}, but the original condition is useful because it leads to easier algebra later.

VanRaden and Freeman (1985) presented formulas which give exact values of accuracy (a_i^2) and prediction error variance (PEV_i) for

certain balanced data sets. In balanced data, accuracies of all elements of u_i are the same, which leads to the identity

$$n_i - trC^*_{ii}k_i = n_i a_i^2 ,$$

where a_i^2 is the squared correlation between an element of u_i and its predictor. Also in balanced data, diagonals of $Z_i'MZ_i$ each equal a constant, say ne_i, the effective number of observations per level of factor i after correcting for fixed effects. Off-diagonals each equal $-ne_i/(n_i-1)$. By substituting these quantities into the expression of VanRaden and Freeman (1985), in the case where there are no factors nesting or nested within the i^{th}, one obtains

$$a_i^2 = ne_i/[ne_i + k_i + ne_i/(n_i-1)].$$

Further algebra gives the effective number of observations per level in balanced data which would provide the same information as that in the unbalanced data.

$$n_i - trC_{ii}k_i = n_i ne_i/[ne_i + k_i + ne_i/(n_i-1)]$$

$$n_i - trC_{ii}k_i = n_i ne_i/[n_i ne_i/(n_i-1) + k_i]$$

$$n_i - trC_{ii}k_i = n_i/[n_i/(n_i-1) + k_i/ne_i]$$

$$n_i/(n_i-1) + k_i/ne_i = n_i/(n_i-trC_{ii}k_i)$$

$$k_i/ne_i = n_i/(n_i - trC_{ii}k_i) - n_i/(n_i - 1)$$

$$ne_i = k_i/[n_i/(n_i - trC_{ii}k_i) - n_i/(n_i - 1)] .$$

The next step is to calculate standard errors as if a simple, unbiased estimation procedure had been used on this imaginary balanced data. One such unbiased estimator is to equate the quadratic $y'MZ_iZ_i'My$ to its expectation, which is

$$E(y'MZ_iZ_i'My) = tr[(MZ_iZ_i'M)Var(y)],$$

$$= tr[MZ_iZ_i'M(I\sigma_0^2 + \sum_j Z_jZ_j'\sigma_j^2)] .$$

By using cyclical commutability and pulling the i^{th} term out of the summation, we get

$$E(y'MZ_iZ_i'My) = tr[Z_i'MZ_i\sigma_0^2 + (Z_i'MZ_i)^2\sigma_i^2$$

$$+ \sum_{j \neq i} Z_i'MZ_jZ_j'MZ_i\sigma_j^2] .$$

This equation is then solved for σ_i^2. This is essentially the MIVQUE (0) estimator, which, in the case of balanced data, gives estimates identical to MIVQUE.

If the j^{th} random effect is cross-classified with the i^{th} and data are balanced, $Z_i'MZ_j$ is null and thus terms involving σ_j^2 drop out of expectations for component i. If the j^{th} random effect is

nested within or a nestor of the i^{th}, the corresponding term involving σ_j^2 remains in the expectation. Ignoring this situation, however, σ_i^2 is estimated as

$$\hat{\sigma}_i^2 = [y'MZ_iZ_i'My - tr(Z_i'MZ_i)\hat{\sigma}_0^2]/tr(Z_i'MZ_i)^2 .$$

Obtaining the variance of this estimator requires evaluating the variance of its quadratic form, which is

$$Var(y'MZ_iZ_iMy) = 2tr[MZ_iZ_i'M(I\sigma_0^2 + Z_iZ_i'\sigma_i^2)]^2 .$$

Expansion of the square in this expression and rearrangement before taking the trace shows that traces of various powers of $Z_i'MZ_i$ are involved. Some patient algebra, again assuming balanced data, produces the relationships

$$tr(Z_i'MZ_i) = ne_in_i ,$$

$$tr(Z_i'MZ_i)^2 = ne_i^2n_i^2/(n_i-1) ,$$

$$tr(Z_i'MZ_i)^3 = ne_i^3n_i^3/(n_i-1)^2 ,$$

and

$$tr(Z_i'MZ_i)^4 = ne_i^4n_i^4/(n_i-1)^3 .$$

With these, the variance of $y'MZ_iZ_i'My$ and consequently of $\hat{\sigma}_i^2$ can

be evaluated, giving

$$\text{Var}(\hat{\sigma}_i^2) = 2\sigma_0^4[n_i - 1 + (n_i - 1)^2/\text{tr}(M^*)]/(ne_i^2 n_i^2)$$

$$+ 4\sigma_0^2\sigma_i^2/(ne_i n_i) + 2\sigma_i^4/(n_i - 1), \qquad [25]$$

where tr(M*) again is the degrees of freedom of the WSS error quadratic. [25] assumes the covariance between $y'MZ_iZ_i'My$ and $\hat{\sigma}_0^2$ is zero, which is the case if σ_0^2 is estimated by WSS (Henderson (1980a)). If an estimator better than WSS is used, $\text{Var}(\hat{\sigma}_i^2)$ obtained from [25] should be an overestimate since it assumes WSS was used. All that is left is substitute estimates of variances for the variances in [25]. [25] appears to give reasonable estimates of standard errors of variances in real data sets but has not been tested with simulation.

Extension of Standard Errors to Relationship Models

For models in which relationships are included, approximate standard errors of $\hat{\sigma}_i^2$ can be obtained in much the same way. The only difference is that, in finding a hypothetical balanced data set with the same amount of information as in the actual data, one must also convert from related animals in the actual data to unrelated animals in the hypothetical data. This is a simple task, though.

The best quadratic for estimating σ_i^2 when relationships are included is $\hat{u}_i'A^{-1}\hat{u}_i$. The expectation of this quadratic is $\text{tr}[A^{-1}(A\sigma_i^2 - C_{ii}\sigma_0^2)]$ or $n_i\sigma_i^2 - \text{tr}(A^{-1}C_{ii})\sigma_0^2$. The corresponding

expectation for unrelated, balanced data would be $n_i\sigma_i^2 - trC^*_{ii}\sigma_0^2$, as before. Derivation of an approximate standard error then proceeds as in the last section with the exception that $tr(A^{-1}C_{ii})$ is computed (or approximated) and is substituted into the formula for ne_i in place of trC_{ii}.

Algorithm for Multi-trait REML which Guarantees Positive Definite Estimates

In preceding sections, each u_i was assumed uncorrelated to all other vectors u_j, $j \neq i$. This is often true with single-trait models, where i and j may represent effects like herds, sires, or interaction. With multiple-trait problems, however, u_i and u_j may represent sire transmitting abilities for different traits and thus a correlation would exist unless traits are genetically uncorrelated. The form of these covariances is usually the same as the variances, so the variance of the adjoined set of vectors $[u_1', u_2', \ldots, u_t']$ can be represented as the Kronecker product of G with either I or A, depending on whether relationships are included. G is the txt matrix of genetic variances and covariances among traits, and t is number of traits. Henderson (1986) has reviewed estimation of genetic parameters in such models.

When X and Z are the same for each trait, the multi-trait mixed model equations can be transformed to a set of t independent single-trait systems (Foulley et al. (1982)). Let the model be

$$y = (I_t \otimes X)b + (I_t \otimes Z)u + e,$$

with $\otimes$ denoting Kronecker product. Vectors y, b, u, and e may each be partitioned into t equal length subvectors representing the t traits. The mixed model equations are

$$\begin{bmatrix} E^{-1} \otimes X'X & E^{-1} \otimes X'Z \\ E^{-1} \otimes Z'X & E^{-1} \otimes Z'Z + G^{-1} \otimes A^{-1} \end{bmatrix} \begin{bmatrix} \hat{b} \\ \hat{u} \end{bmatrix} = \begin{bmatrix} (E^{-1} \otimes X')y \\ (E^{-1} \otimes Z')y \end{bmatrix},$$

where E is the txt variance-covariance matrix among errors. Absorption of equations for b into equations for u is not difficult, giving

$$[E^{-1} \otimes Z'MZ + G^{-1} \otimes A^{-1}]\hat{u} = (E^{-1} \otimes Z'M)y,$$

with M defined as before. Transformation to independent single-trait equations is accomplished by choosing a rotation P^{-1} such that G^{-1} and E^{-1} are both diagonalized, or equivalently that G and E are both diagonalized. The transformation can be applied by creating y*, where $y^* = (P^{-1} \otimes I)y$.

The previous development is as if G and E are known matrices. Obviously, if variances are to be estimated, this information is not known. Consequently, the rotation P^{-1} is based on priors for G and E, and REML estimation can proceed as follows.

A set of t new traits are chosen as linear combinations of the original traits. If the prior was correct, these t new traits would produce a set of $\hat{u}_i$'s uncorrelated with each other. Let U represent the adjoined set of vectors $[\hat{u}_1, \hat{u}_2, \ldots, \hat{u}_t]$, where the adjoining is now side-by-side rather than end-to-end and each $\hat{u}_i$ represents a solution vector for one of the rotated traits. The quadratics U'U

or $U'A^{-1}U$ are assumed positive definite. This assumption is reasonable if all original traits were measured traits, meaning that no original trait was created simply as a linear combination of other measured traits. The assumption of positive definiteness must be true if convergence is to occur, because convergence requires that each $\hat{u}_i$ be not only linearly independent but also orthogonal to the others.

The genetic variance-covariance matrix among rotated traits (G*) can be estimated by $VU'A^{-1}UV$ or $VU'UV$, where V is a diagonal matrix whose diagonals are $1/\sqrt{n_i - tr(A^{-1}C_{ii})k_i^*}$ if the model contains relationships or $1/\sqrt{n_i - trC_{ii}k_i^*}$ if the model is without relationships. k_i^* was used in place of k_i to clarify that these are ratios for the rotated traits rather than original traits. The expression $VU'UV$ simplifies to [19] if only a single trait is considered. One does not wish to estimate G*, however, but G. This is done by estimating G* and then using the reverse rotation to get back to G. Let the original transformation matrix be labeled P^{-1}. It was chosen in such a manner that $P'^{-1}GP^{-1}$ would be diagonal if the prior for G was correct. The reverse rotation then consists of estimating G by

$G = P'G^*P$, which is

$G = P'VU'UVP$ in the no-relationship model and

$G = P'VU'A^{-1}UVP$ in the relationship model.

Because A^{-1}, U, V, and P are all of full column rank, the estimate of G is positive definite.

Efficiency of Multi-trait REML with Rotation Compared to Single-trait Methods

Estimation of genetic variance and covariance components by multi-trait REML with rotation requires that, for each round of estimation, t solution vectors and traces of inverses be computed, where t again is the number of traits. If direct inversion is used, computation of trC_{ii} represents most of the work for each trait. If bounds procedures are used, computing $\hat{u}_i$'s requires the most work. If diagonalization of Z'MZ is used, additional traits or additional rounds of iteration are almost no cost compared to the one-time initial cost of diagonalization.

Several other items besides those just mentioned must be calculated in each iteration, but these require little computing time. Such items include inverting, multiplying, and obtaining eigenvectors for matrices of size txt and rotating the right-hand sides by post-multiplying a matrix of size txt. All that is required to obtain covariance estimates in each round is to multiply solution vectors together to form $\hat{u}_i'\hat{u}_j$ or $\hat{u}_i'A^{-1}\hat{u}_j$, neither of which are expensive. Thus, cost of estimating both variances and covariances by multi-trait REML with rotation is about the same as t single-trait analyses for variances only.

Another method for estimating genetic variances and covariances

is discussed by Schaeffer (1983). Variances are estimated by t single-trait analyses. Covariances are estimated by single-trait procedures also, but for these the sum of each possible pair of traits is treated as a new trait and its genetic variance is estimated. Then the genetic covariance is obtained with the relationship

$$\mathrm{Var}(y_1 + y_2) = \mathrm{Var}(y_1) + \mathrm{Var}(y_2) + 2\mathrm{Cov}(y_1, y_2) .$$

A total of t(t+1)/2 single-trait analyses are required for Schaeffer's procedure compared to just t analyses using multi-trait REML with rotation. If convergence rates are similar, this could represent a considerable savings by using multi-trait REML. One situation when convergence rates might be different is when certain covariances are trying to go outside the parameter space. Multi-trait REML may not converge in this case but only slowly approach the parameter boundary, whereas the procedure of Schaeffer would simply give estimated covariances outside the parameter space. Despite possible slower convergence, multi-trait REML is still preferred because most users would like to have positive definite estimated G's and because true REML requires that estimates be within the parameter space.

Estimation of Accuracies and Prediction Error Variances in Relationship Models

Accuracies (a^2) and prediction error variances (PEV) for breeding value estimates are two items usually desired by animal breeders when genetic evaluations are done. Accuracies are squared correlations between elements of u and $\hat{u}$ while PEV are the expected squared differences between elements of u and $\hat{u}$. These measures allow the user to determine how reliable or how much potential change there might be for a particular animal's evaluation. VanRaden and Freeman (1985) presented strategies to obtain very close upper and lower bounds on these quantities in models not containing relationships. In this section, strategies for approximating a^2 and PEV in models with relationships are presented.

PEV and a^2 are both simple functions of diagonals of C_{ii}, the appropriate segment of the inverse of (Z'MZ + K). A simple method for estimating diagonals of an inverse is to take reciprocals of diagonals of the original matrix. This is used to obtain PEV in, for instance, the NAAB calving ease evaluation (Berger, P. J., personal communication (1985)). This method always underestimates PEV and overestimates a^2, as can be proved by algebra similar to that in VanRaden and Freeman (1985).

A particular animal's PEV or a^2, if estimated by reciprocal of the diagonal, is estimated as if all its relatives' breeding values were known without error. This is demonstrated as follows. Suppose there exist a bull and his sire having n_B and n_S progeny, respectively.

Suppose also that this bull and his sire are the only random factors and that fixed effect constants are known and have been subtracted from the data. BLUP predictors of transmitting abilities for the bull (u_B) and his sire (u_S) using relationships would be obtained from equations

$$\begin{bmatrix} n_B + k4/3 & -k2/3 \\ -k2/3 & n_S + k4/3 \end{bmatrix} \begin{bmatrix} \hat{u}_B \\ \hat{u}_S \end{bmatrix} = \begin{bmatrix} y_{B.} \\ y_{S.} \end{bmatrix} .$$

PEV and a^2 for the bull would be obtained from the first diagonal of the inverse of the above 2x2 matrix. Let this diagonal be represented by x. Then $x = (n_S + k4/3)/(n_B n_S + (n_B+n_S)k4/3 + k^2 4/3)$. Of interest is the value of x as n_S goes to infinity, which is equivalent of saying that the sire's transmitting ability becomes known completely. Taking derivatives of numerator and denominator with respect to n_S gives

$$x \to 1/(n_B + k4/3) \quad \text{as} \quad n_S \to \infty .$$

In other words, x approaches the reciprocal of the diagonal as the amount of information on the sire grows large. In general, then, $A^{-1}k$ adds to the diagonal a factor which acts as if all relatives had perfect information.

Another strategy to estimate PEV or a^2 would be to ignore information coming through A^{-1} and to use only information on the individual

himself (actually on the individual's progeny in a sire model). In the example just given, this would mean using $1/(n_B + k)$ instead of $1/(n_B + k4/3)$ as the estimate of x. This would be an appropriate strategy if relatives had little or no information themselves. The estimate $1/(n_B + k4/3)$ would always underestimate x and $1/(n_B + k)$ would usually, but not always, overestimate x.

An estimate somewhere between these two extremes would be desirable. The nonzero off-diagonals in each row of A^{-1} can be thought of as indicators of relatives' direct contributions to accuracy of the animal in question. One needs only to know how much information they actually contribute. A useful measure of information is a^2, because it combines additively among relatives, whereas its square root, a, would not. For example, an animal with no information itself would have a^2 equal to 1/4 times its sire's a^2 plus 1/16 times its maternal grandsire's (MGS) a^2, if these were the ancestors used. Another property of a^2 is that it varies within the range 0 to 1, which makes it useful for interpolating between the two simple estimates of the previous paragraph.

Another problem is how to combine a^2 of different relatives into a single factor. Because off-diagonals of A^{-1} can be either positive or negative, obvious choices would be either absolute values or squares of off-diagonals of A^{-1} as weights for relatives' a^2's. Use of squares has more theoretical appeal, as can be demonstrated with the example of the previous paragraph. Off-diagonals of A^{-1} for

sire and MGS would be -8/11 and -4/11. Factors to calculate an individual's a^2 from sire and MGS were 1/4 and 1/16, respectively. To obtain relative weights with four times as much emphasis on sire as on MGS, squares of off-diagonals of A^{-1} should be used rather than absolute values.

A remaining question is what estimates one should use for a^2 of the relatives. An iterative strategy might work in which an individual's a^2 is calculated as a function of its relatives and vice versa until convergence. Such a strategy might tend to over-estimate true a^2, however, because, for instance, the information that a sire passes on to a son would be reflected back and show up as information on the son contributing to sire a^2. This problem might be avoided by estimating a^2 of relatives as if their own information was the only information contributing to their a^2. This would mean using $ne_i/(ne_i + k)$ to estimate a^2 of the i^{th} relative, where ne_i is the diagonal of Z'MZ corresponding to the i^{th} relative.

The proposed estimator is to first calculate the sum of squared off-diagonals in a row of A^{-1} multiplied by the corresponding simple estimates of a^2 just described, all divided by the sum of squared off-diagonals of A^{-1}. Call this w. The corresponding diagonal of the inverse is then estimated as $1/[ne_i + w\ diag(A^{-1})k + (1-w)k)$, which is just an interpolation between the two earlier simple estimators. PEV and a^2 can be estimated directly from this estimate of the diagonal.

RESULTS

Simulated Data

Simulated data sets were generated to study performance of the bounds on REML in various situations. A primary interest was to compare the bounds procedure to another quick method of obtaining approximate REML solutions, that being iterated Henderson's simple method (IHSM). IHSM has not been studied in great detail before but is a natural extension of a procedure presented by Henderson (1980a). The idea is that if Henderson's simple method (HSM) is approximately MIVQUE, and iterated MIVQUE is REML, then IHSM should be approximately REML. Later results will show this logic is reasonable.

The first model studied contained only herds and sires in an unbalanced pattern resembling dairy field data. Different ratios of error to sire variance (k) were studied, but only k = 15 will be presented. Results from 25 trials of variance component estimation on data sets containing 400 herds and 100 sires are in Table 1. Conclusions from these results are that fairly tight bounds on REML can be produced and that the average of upper and lower bounds behaves nearly the same as IHSM. About half the time IHSM produced estimates outside the bounds, i.e., farther away from the true REML estimate than the bounds.

The next model was identical to the first except that the data contained selection. Selection was introduced by generating an original random sample of progeny on sires and then generating

Table 1. Results from herd and sire model comparing bounds on REML with iterated Henderson's simple method[a]

$\hat{\sigma}^2_1$	Upper	Lower	(U+L)/2	IHSM[b]
mean	.972	.955	.964	.968
s.d.	.193	.189	.191	.190

[a]Estimation of $\sigma^2_1 = 1.0$, 25 trials.

[b]In 13 of 25 trials IHSM was outside the bounds.

additional progeny for only those 20 sires that ranked best based on the first set. Such selection is of the L'y type described by Henderson (1980b). Results from 25 trials are in Table 2. Bounds were apparently unaffected by selection whereas IHSM showed substantial downward bias. In none of the 25 trials was IHSM inside the bounds.

Table 2. Results from data with selection using a herd and sire model[a]

$\hat{\sigma}^2_1$	Upper	Lower	(U+L)/2	IHSM[b]
mean	.991	.973	.982	.527
s.d.	.209	.205	.207	.130

[a]Estimation of $\sigma^2_1 = 1.0$, 25 trials.

[b]In 25 of 25 trials IHSM was outside the bounds.

Three other models were studied, all of these without selection. Tables 3, 4, and 5 give results for the following models: a model with herds, sires, and interaction; a model with two cross-classified random effects; and a multi-trait model with both variance and covariance components to estimate. Size of data sets was reduced to 20 sires and 40 herds for these three models to speed computation. In all cases bounds were narrow enough to be useful, and in all cases averages of the two bounds gave performance similar to IHSM. Finally, correlations of estimates produced by IHSM with the average of bounds on REML were in the range of .90 to .99, indicating that the two procedures utilize information in about the same way.

Table 3. Results from model with herds, sires, and interaction[a]

$\hat{\sigma}_1^2$	Upper	Lower	(U+L)/2	IHSM[b]
mean	.940	.900	.920	.954
s.d.	.344	.326	.335	.348
$\hat{\sigma}_2^2$				
mean	1.110	.900	1.005	1.030
s.d.	.389	.329	.359	.414

[a]Estimation of $\sigma_1^2 = 1.0$, $\sigma_2^2 = 1.0$, 25 trials.

[b]IHSM was outside the bounds in 18 of 25 trials for component 1 (sire variance), 7 of 25 for component 2 (interaction variance).

Table 4. Results from cross-classified random model[a]

$\hat{\sigma}_1^2$	Upper	Lower	(U+L)/2	IHSM[b]
mean	1.011	.920	.965	.987
s.d.	.315	.306	.310	.320
$\hat{\sigma}_2^2$				
mean	1.033	.947	.990	.996
s.d.	.380	.360	.370	.340

[a]Estimation of $\sigma_1^2 = 1.0$, $\sigma_2^2 = 1.0$, 25 trials.

[b]IHSM was outside the bounds in 18 of 25 trials for component 1, 18 of 25 for component 2.

Table 5. Results from multi-trait model[a]

$\hat{\sigma}_1^2$	(U+L)/2[b]	REML	IHSM
mean	.979	1.000	.990
s.d.	.277	.284	.284
$\hat{\sigma}_2^2$			
mean	1.033	1.054	1.057
s.d.	.423	.433	.446

[a]Estimation of $\sigma_1^2 = 1.0$, $\sigma_2^2 = 1.0$, $\sigma_{12} = .2$, 25 trials.

[b]Average of upper and lower bounds.

Table 5. (continued)

$\hat{\sigma}_{12}$	$(U+L)/2^b$	REML	IHSM
mean	.287	.292	.286
s.d.	.261	.267	.263

Real Data

Simulated data works well for comparing methods of estimation under certain easy to program circumstances, but one also needs to be sure such methods give efficient and useful estimates of variances when applied to real data problems. Four real data sets were studied, all of which might be considered of moderate size by animal breeding standards. Number of animals ranged from 1237 to 49,918, number of sires from 61 to 428, and number of traits from 1 to 24. REML estimates for all four problems were obtained with a program which handles either single-trait or multiple-trait problems, either related or unrelated sires, and computes estimates with either bounds procedures or diagonalization of Z'MZ and A^{-1}. This program, as well as an absorption program to create the needed matrices and some documentation, is in the Appendix.

The first data set was from William Foster and was part of his Master's thesis project (Foster (1985)). Data were scores on 16 linear type traits from 21st Century Genetics Cooperative, Shawano, WI.

Additionally, milk and fat production records from DHIA were merged and edited so that each remaining record contained all of the 18 traits. Editing was also for sire so that only the 228 sires with at least 20 progeny were included in the analysis. Usable records were 43,428 from 5171 herd-years, with less than 3 percent of sire by herd-year subclasses filled. Sires were assumed unrelated.

Approximate multi-trait REML estimates were obtained by iterating on the average of upper and lower bound formulas until convergence. One additional round was completed with exact expectations obtained by inversion to compare costs and to determine how closely bounds had converged to exact REML. Cost with bounds was $3.50 per round whereas cost of the final iterate with inversion was $185. Foster (1985) obtained Method III estimates as well by using the RANDOM option of GLM of SAS (SAS (1985)). The cost of these was not documented. Finally, eigenvectors of Z'MZ were obtained and REML estimates computed on the diagonalized system. Cost of estimation with this procedure was $38, including $25 for absorption.

Heritability estimates from these four procedures and also from single-trait REML are in Table 6. Estimates from averages of bounds are very close to but always higher than exact REML estimates, indicating merely that the bounds were not centered about the exact value of the trace. Single-trait solutions differed slightly from multi-trait solutions and Method III estimates differed substantially from either of these. No practical differences were observed between estimates

Table 6. Heritability estimates from Foster's data by five methods

Trait	Method III	Multiple Trait			
		Bounds[a]	REML1[b]	REML2[c]	Single[d]
Basic form	.37	.449	.444	.443	.441
Strength	.18	.258	.255	.255	.261
Dairyness	.18	.276	.273	.272	.271
Stature	.33	.367	.363	.362	.362
Body depth	.38	.324	.320	.319	.322
Rump angle	.21	.248	.245	.244	.245
Legs (side v.)	.19	.173	.171	.170	.174
Foot angle	.12	.099	.098	.098	.100
Fore udder	.17	.195	.193	.193	.191
Udder depth	.22	.264	.261	.259	.264
Rump width	.25	.182	.180	.180	.185
Legs (rear v.)	.09	.094	.093	.093	.094
Rear ud. height	.16	.207	.205	.204	.204
Rear ud. width	.14	.159	.158	.157	.153
Susp. lig.	.15	.141	.140	.138	.139
Teat. pl.	.21	.202	.200	.200	.200
MEMILK		.160	.158	.158	.147
MEFAT		.180	.178	.177	.175

[a]Approximate REML using averages of bound formulas.

[b]One iterate with inversion using converged bounds solutions as priors.

[c]REML solutions using diagonalized Z'MZ.

[d]Single-trait REML solutions.

by any of the REML algorithms. Clearly, estimates of variances and covariances with very desirable properties can be obtained from data sets of this size fairly cheaply despite the large number of traits. Both bounds procedures and diagonalization were effective in reducing costs, with diagonalization cheaper in this situation. It should be remembered that costs of diagonalization rise in proportion to number of sires cubed rather than number of sires squared for bounds. Conversely, costs of bounds rise in direct proportion to number of traits whereas many additional traits can be handled for very little cost once diagonalization has been done.

The second data set was from Mounir Djemali and was used to examine properties of an ordered categorical model for sire evaluation. Only one trait was analyzed, dystocia, scored on a discrete scale from one to five. Number of records and sires were 5026 and 225, respectively. Further description of data and editing is in Djemali (1985). My purpose was to obtain a REML estimate of heritability of dystocia as scored to compare with estimated heritability of an underlying normal variate from the ordered categorical approach.

From ordered categorical procedures, heritability for the underlying variate was estimated to be .148 with relationships and .147 without (Djemali (1985)). For dystocia analyzed as scored with a mixed linear model, exact REML estimates were .0548 with relationships and .0553 without. These lower estimates were expected because heritability of the observed variate must necessarily be smaller than

that of the underlying variate. Bounds procedures gave estimates of .0696 with relationships and .0552 without.

Two things are apparent from these results. First, including relationships in the model did not change the estimate of heritability very much. Second, bounds procedures worked exceptionally well in the no-relationship case but less satisfactorily when relationships were included. Costs of obtaining the four estimates and the estimates are in Table 7. For this problem, bounds were less expensive than diagonalization.

Table 7. Heritability estimates and costs of estimates for dystocia data

	Type of Estimation	
	Bounds	Diagonalization
Relationships:		
not included	.0552, $4	.0553, $10
included	.0696, $7	.0548, $31

The third data set was from Lyons et al. (1986). It contained 22 measures of health, most of which were binomial, plus milk and fat production on 1237 cows representing 61 sires. Sires were assumed unrelated. The expected low heritabilities of several traits, the large number of traits, and the small amount of data virtually ensured that the program would not converge to a positive definite matrix but rather would

attempt to produce a positive semi-definite matrix as an estimate of genetic variances and covariances. Positive semi-definite matrices, however, do not lend themselves well to computation and can cause problems with numerical stability.

Numerical problems surfaced in this data set in the form of an attempt to find the square root of a negative number in the seventh round of iteration. Problems of this nature might occur often in multiple-trait estimation problems, so changes were made to the program to allow convergence to a point close to the border of the parameter space but far enough away to prevent numerical instability. The method to accomplish this was to apply constraints to the eigenvalues of $T^{-1}GT'^{-1}$, where T is chosen such that $TT' = E$. Eigenvalues of $T^{-1}GT'^{-1}$ were set equal to 10^{-5} if they fell below this magnitude.

Constraining eigenvalues to remain positive ensures that heritabilities of all traits will remain far enough from zero that the algorithm will not fail. It is usually also desirable to constrain heritabilities to be 1.0 or less. This can be accomplished at the same time by setting eigenvalues equal to 1/3 if they rise above 1/3. Reciprocals of eigenvalues of $T^{-1}GT'^{-1}$ are the ratios of σ_0^2/σ_1^2 for the rotated traits. Thus, eigenvalues of 1/3 or values for σ_0^2/σ_1^2 of three correspond to heritabilities of one for sire models. Use of these constraints forces heritabilities of not only the original traits but of any linear combination of traits to stay between zero and one.

In the health traits analysis, 13 of 24 eigenvalues had to be forced to remain above 10^{-5}, indicating that the estimated genetic variance-covariance matrix would have rank of only 11. Additionally, one eigenvalue had to be limited to 1/3, meaning that the heritability of one linear combination of traits attempted to rise above one. Heritability estimates for the original traits are in Table 8.

Table 8. Heritability estimates for 22 measures of health and for milk and fat production

Trait	h^2	Trait	h^2
sold open	.10	displaced abomasum	.13
abortion	.06	other digestive	.06
cystic ovaries	.19	trim feet	.29
retained placenta	.22	leg problems	.09
uterine infection	.24	foot problems	.41
no. of inseminations	.06	crampy	.14
other reproductive	.05	other locomotive	.08
mastitis	.23	pneumonia	.37
udder injury	.17	other respiratory	.15
other udder	.02	all other	.05
milk fever	.45	milk	.18
ketosis	.23	fat	.22

It was surprising that none of the original traits had heritabilities lower than .018 since 13 of the rotated traits did. A few traits, for instance milk fever, had heritabilities much higher than expected.

It should be remembered, however, that REML does not give unbiased estimates. Heritabilities are likely to be upwardly biased in this data set because of the large number of eigenvalues constrained to be positive.

The fourth data set was from the dissertation research of Les Hansen (Hansen (1981)). These data had been analyzed previously with Method III and also with single-trait REML (without relationships) for just a few variances but no covariances. New computing strategies made it cost-effective to reanalyze these data using multiple-trait REML with relationships.

Complete information on production and reproduction was available for 49,918 first lactation cows. A subset, 15,684 cows, also had information on their reproductive traits as heifers. Four hundred twenty-eight sires were included in the analysis, 353 of these because they had at least 10 daughters in the data and the remaining 75 because they were necessary for construction of A^{-1}. Although Hansen (1981) analyzed 21 traits, several of these were linearly confounded or nearly so, and this would give singular matrices and computing problems. An example of singularity would be to include milk, fat, and fat corrected milk in the same analysis. Thus, no more than nine traits were analyzed in any one model.

Again, an interest was to compare estimates of heritability with relationships either included or excluded. Inclusion of relationships gave heritability estimates slightly larger in four of five traits

and barely smaller in the fifth. Estimates are listed in Table 9.

Table 9. Estimates of heritabilities for production and reproduction with and without relationships in the model

Trait[a]	Relationships not included	Relationships included
FCM120	.167	.183
FCM305	.204	.225
FB150	.030	.032
NS3	.017	.017
D0150	.031	.032

[a]FCM120 - 120 day fat-corrected milk production; FCM305 - 305 day fat-corrected milk production; FB150 - days to first breeding, maximum 150; NS3 - number of services, maximum 3; D0150 - day open, maximum 150.

Another interest was in cost. Absorption of HYS for the 49,918 records cost $10. Twenty iterations of the REML algorithm using bounds (without relationships) for a 5-trait model cost $57 but did not quite produce convergence. Simultaneous diagonalization of $Z'MZ$ and A^{-1} followed by 27 iterations for a 7-trait model produced convergence and cost $190. In this last model, two eigenvalues had to be held away from zero.

Programs for Estimating G and E

Two programs were developed which can be used in tandem to obtain multiple-trait REML estimates of G and E for a variety of models commonly used by animal breeders. The first program, ABSORBM, creates and outputs matrices for the second program, REMLM (M stands for multiple-trait). The programs are in Fortran and use International Mathematical and Statistical Libraries, Inc. (IMSL) subroutines. The programs are written to allow easy comparisons of different models applied to the same data set.

Most of the computing strategies discussed in the Methods Section have been incorporated into program REMLM. One which was not was algorithm [22], because it does not adapt as easily to multi-variate REML. Perhaps the biggest limitation of the programs are that they do not allow for other random factors in the model besides sire and error. This was necessary because the addition of other random factors to a multi-trait model would require that three variance matrices be simultaneously diagonalized, and this cannot generally be done. ABSORBM does allow herd-year-seasons to be declared random for single-trait models, but herd-year-season variance is not estimated but rather a value supplied by the user is used as if known.

Size of data sets handled by the programs is limited only by the number of sires (up to approximately 800 for the NAS AS/9160). Virtually any number of records, any number of levels of fixed effects, and any number of traits can be accommodated. Problems with specifying positive

definite priors for G and E when these are large matrices are alleviated by having the computer form priors from the data. A positive definite prior for E is obtained from the WSS quadratic. A prior for G is then obtained by dividing the prior for E by a scalar chosen by the user. This procedure assumes that heritabilities of all traits are equal and that genetic correlations equal error correlations.

Convergence is declared when all variances differ by no more than .001 of their magnitudes and no correlations differ by more than .001 absolutely from one round to the next. The user may also force termination after any number of rounds. REMLM outputs converged values of variances, covariances, correlations, heritabilities, solutions for genetic groups, and estimated transmitting abilities of sires if desired. Approximate standard errors are also output where feasible. Much more work is needed on methods for computing standard errors of REML estimates of variance components.

Estimates of Diagonals of Inverse in Relationship Models

Strategies presented in the methods section for estimating diagonals of $(Z'MZ + A^{-1}k)^{-1}$ were applied to the dystocia data of Djemali (1985). Results are given in terms of squared accuracy (a_i^2) which is a direct function of these diagonals. Three levels of k were examined, k = 79, 15, and 7, corresponding to heritabilities of .05, .25, and .50. Effective numbers of progeny (ne_i) from diagonals of Z'MZ for these data ranged from 0 to 97 for the 228 sires. Sires with $ne_i = 0$ were

those included to allow for easy construction of A^{-1}. For easy reference, strategies are named and described in Table 10.

Table 10. Strategies used to estimate squared accuracies of sire estimated transmitting abilities

Name	Formula	Description
S1	$1/(ne_i + diag_i(A^{-1})k)$	reciprocal of diagonal
S2	$1/(ne_i + k)$	reciprocal of diagonal as if animal had no relatives
S3	$1/(ne_i$ + adjustment)	interpolation between S1 and S2, using S2 to estimate a_j^2 of relatives
S4	$1/(ne_i$ + adjustment*)	interpolation between S1 and S2, using iteration to obtain a_j^2 of relatives
S5	NxN inversion	direct inversion of matrix for the N closest relatives
S6	(S3 + S5)/2	average of best two strategies

Means of estimated a_i^2 for the six strategies and for the three levels of k are in Table 11, along with means of true a_i^2 (obtained by inverting the sire coefficient matrix) and correlations of estimated with true. For all six strategies, estimates of a_i^2 were better, both in terms of mean and correlation with true a_i^2, for smaller values of k (larger heritabilities). This is explained by the fact that, as k decreases, Z'MZ becomes larger relative to $A^{-1}k$. When one compares

Table 11. Estimation of squared accuracies by six strategies for sires with relationships

	k = 79		k = 15		k = 7	
Strategy	Mean a_i^2	Corr.[a] $\hat{a}_i^2, a_i^2$	Mean a_i^2	Corr. $\hat{a}_i^2, a_i^2$	Mean a_i^2	Corr. $\hat{a}_i^2, a_i^2$
True	.244	-	.514	-	.638	-
S1	.400	.755	.583	.965	.683	.990
S2	.175	.824	.455	.892	.589	.908
S3	.218	.946	.518	.992	.652	.998
S4	.302	.966	.548	.996	.665	.998
S5	.313	.957	.543	.997	.661	.999
S6	.266	.979	.530	.997	.657	.999

[a]Correlation of estimated a_i^2 with true a_i^2.

A^{-1} and Z'MZ, one finds that off-diagonals of Z'MZ are in general smaller relative to diagonals than those of A^{-1}. Thus, as the influence of A^{-1} decreases, diagonals of $(Z'MZ + A^{-1}k)^{-1}$ become easier to estimate.

Of the two simpler strategies, S1 outperformed S2 for k = 7 but the reverse was true for k = 79. This might be expected based strictly on average a_j^2 of relatives. S1 assumes that all relatives have $a_j^2 = 1$ (complete information), whereas S2 assumes all relatives

have $a_j^2 = 0$ (no information). As k increases, average a_j^2 decreases, which makes the assumption of S2 closer to the truth.

The remaining strategies (S3-S6) each outperformed both of the simple strategies across all levels of k. S1 and S5 both give upper bounds on a_i^2, but S5 contains more information, so it gives a lower upper bound. Of strategies S3, S4, and S5, none was universally superior to the others. S3 had mean estimated a_i^2 closer to the true mean than S4 or S5, but the correlation of estimated with true was not as high for S3 as for S4 or S5. The average of S3 and S5, denoted S6, gave very good performance both in terms of mean and correlation for all levels of k and was the best strategy overall. S6 requires more work to compute, however, because it involves two different strategies and this somewhat offsets its advantages.

SUMMARY AND CONCLUSIONS

Several strategies were presented which may be helpful for estimating variances and covariances in mixed models. Formulas were presented which allow exact upper and lower bounds to be calculated on traces of inverses for models in which animals are assumed unrelated. These can be used to fairly cheaply obtain upper and lower bounds on REML estimates of variance components or approximate REML estimates using the average of the upper and lower bound formulas. These procedures were applied to simulated data from five different models and gave satisfactory results in each case.

Three algorithms for computing REML estimates of variances were discussed. Two of these are already well-known. Of these two, one [19] was shown to always be superior to the other [18] based on arguments from balanced data. The third was derived to give convergence nearly as rapidly as iterative MIVQUE at a fraction of the cost.

An algorithm was developed which guarantees positive definite estimated variance-covariance matrices in multiple-trait problems. If eigenvalues are constrained, this algorithm can converge to a point arbitrarily close to the edge of the parameter space (yielding an "almost singular" matrix) without encountering numerical problems. Similarly, eigenvalues can be constrained such that no trait and no linear combination of traits has an estimated heritability greater than one. The algorithm requires that all traits be measured on all

animals. It is based on a simultaneous diagonalization of E and G, which are the txt error and genetic variance-covariance matrices (t is number of traits). This algorithm produces multi-trait REML estimates of variances and covariances for nearly the same cost as would be required to estimate variances only using single-trait REML.

The same procedures used to diagonalize E and G can also be used to simultaneously diagonalize Z'MZ and A^{-1} if this is desired. This strategy is cost-effective in small to moderate sized problems or in data sets having a large number of traits. Initially, diagonalization of Z'MZ and A^{-1} is expensive, but is cost effective because it only has to be performed once versus once for each trait and for each iterate as with bounds. Procedures to obtain approximate traces in models which include relationships were developed. These procedures may be useful but are not as pleasing as the bounds in no-relationship models. Strategies to estimate accuracies and prediction error variances in relationship models were also presented and some of these gave very satisfactory results in real data situations.

Finally, a general-purpose Fortran program was developed for purposes of estimating variances and covariances in small to moderate-sized problems. The program handles single- or multiple-trait problems, related or unrelated sires, genetic groups or no genetic groups, and computes with either an exact procedure (diagonalization) or approximate procedures (estimation of traces). Estimates of error variances, sire variances, heritabilities, standard errors of these,

plus estimates of genetic, phenotypic, and error correlations and covariances among all traits are output. Additionally, estimates of sire transmitting abilities can be output. These are computed by using estimated E and G matrices in place of true E and G.

The program was applied to four data sets made available by colleagues. This was done both to get a better feel for the properties of multi-trait REML estimation in real data and simply because the estimates of heritabilities and genetic correlations which were produced were needed. Interesting findings were that 1) convergence occurred within a reasonable number of iterations (less than 30) for all problems, 2) including relationships in the models caused modest increases in heritability estimates for production traits but smaller changes in estimates for reproduction and dystocia, and 3) estimated variance matrices often approached singularity and had to be constrained to avoid numerical instability.

Few, if any, methods of estimating variances and covariances exist which have more intuitive or theoretic appeal than restricted maximum likelihood. Until recently, however, REML was used infrequently because of computing difficulties. The programs of the Appendix should help make REML estimates easier to obtain and more affordable in at least reasonably sized problems and for a limited variety of models. For larger data sets or more complex models, bounds procedures and other approximate procedures may prove helpful. REML estimation of variances and covariances is not always easy, but is often worthwhile.

REFERENCES CITED

Dempfle, L., C. Hagger, and M. Schneeberger. 1983. On the estimation of genetic parameters via variance components. Génét. Sél. Evol. 15:425.

Dempster, A. P., C. M. Patel, M. R. Selwyn, and A. J. Roth. 1984. Statistical and computational aspects of mixed model analysis. Appl. Stat. 33:203.

Djemali, M. 1985. The economic importance of dystocia in diary cattle and development of an ordered categorical analysis procedure to evaluate sires for calving ease. Unpublished Ph.D. Thesis, Iowa State University, Ames, Iowa.

Forsythe, G. E., M. A. Malcolm, and C. B. Molar. 1977. Computer methods for mathematical computations. Prentice-Hall, Inc., Englewood Cliffs, New Jersey.

Foster, W. W. 1985. Effects of first lactation linear type scores on first lactation production and herdlife of Holstein dairy cattle. Unpublished M.S. Thesis, Iowa State University, Ames, Iowa.

Foulley, J. L., S. Calomiti, and D. Gianola. 1982. Ecriture des équations du BLUP multicacteres. Ann. Génét. Sél. Animal 14:309.

Freeman, A. E. 1979. Components of variance: Their history, use, and problems in animal breeding. In L. D. VanVleck and S. R. Searle (eds.). Variance components and animal breeding. Cornell University.

Gianola, D. and R. L. Fernando. 1986. Bayesian methods in animal breeding theory. J. Anim. Sci. 63:217.

Graybill, F. A. 1983. Introduction to matrices with applications in statistics. 2nd ed. Wadsworth Publishing Company, Inc., Belmont, California.

Hansen, L. B. 1981. Genetic relationship of yield and fertility in dairy cattle. Unpublished Ph.D. Dissertation, Iowa State University, Ames, Iowa.

Harville, D. A. 1977. Maximum likelihood approaches to variance component estimation and to related problems. J. Amer. Statist. Assoc. 72:320.

Henderson, C. R. 1953. Estimation of variance and covariance components. Biometrics 9:226.

Henderson, C. R. 1975. A rapid method for computing the inverse of a relationship matrix. J. Dairy Sci. 58:1727.

Henderson, C. R. 1980a. A simple method for unbiased estimation of variance components in the mixed model. 72nd Annual Mtg. Amer. Soc. Anim. Sci., Cornell University, Ithaca, NY.

Henderson, C. R. 1980b. Unbiased estimation of variance components under a selection model. J. Dairy Sci. 63(Suppl. 1):110. (Abstr.)

Henderson, C. R. 1986. Recent developments in variance and covariance estimation. J. Anim. Sci. 63:208.

Hohn, F. E. 1973. Elementary matrix algebra. 3rd ed. The Macmillan Company, NY.

Hudson, G.F.S. and L. D. VanVleck. 1982. Estimation of components of variance by method 3 and Henderson's new method. J. Dairy Sci. 65:435.

LaMotte, L. R. 1970. A class of estimators of variance components. Technical Report No. 10. Department of Statistics, University of Kentucky, Lexington.

LaMotte, L. R. 1971. Locally best quadratic estimators of variance components. Technical Report No. 22. Department of Statistics, University of Kentucky, Lexington.

LaMotte, L. R. 1973. Quadratic estimation of variance components. Biometrics 29:311.

Lin, C. Y. and A. J. McAllister. 1984. Monte Carlo comparison of four methods for estimation of genetic parameters in the univariate case. J. Dairy Sci. 67:2389.

Lyons, D. T., A. E. Freeman, and P. J. Berger. 1986. Evaluation of health-related traits in Holstein cattle. J. Dairy Sci. 69(Suppl. 1):124. (Abstr.)

Meyer, K. 1985. Maximum likelihood estimation of variance components for a multivariate mixed model with equal design matrices. Biometrics 41:153.

Patterson, H. D. and R. Thompson. 1971. Recovery of interblock information when block sizes are unequal. Biometrika 58:545.

Rao, C. R. 1971a. Estimation of variance and covariance components - MINQUE theory. J. Multivariate Analysis 1:257.

Rao, C. R. 1971b. Minimum variance quadratic unbiased estimation of variance components. J. Multivariate Analysis 1:445.

Rao, C. R. 1972. Estimation of variance and covariance components in linear models. J. Amer. Statist. Assoc. 65:161.

SAS Institute, Inc. 1985. SAS User's Guide: Statistics, 1985 edition. SAS Institute, Inc., Cary, NC.

Schaeffer, L. R. 1983. Notes on linear model theory, best linear unbiased prediction, and variance component estimation. University of Guelph, Guelph, Ontario.

Schaeffer, L. R. 1986. Estimation of variances and covariances within the allowable parameter space. J. Dairy Sci. 69:187.

Searle, S. R. 1968. Another look at Henderson's methods of estimating variance components. Biometrics 24:749.

Skaar, B. R. 1985. Direct genetic and maternal variance and covariance component estimates from Angus and Hereford field data. Unpublished Ph.D. Dissertation, Iowa State University, Ames, Iowa.

Taylor, J. F., B. Bean, C. E. Marshall, and J. J. Sullivan. 1983. Performance of the EM algorithm for multiple-trait REML. J. Dairy Sci. 66(Suppl. 1):124. (Abstr.)

VanRaden, P. M. and A. E. Freeman. 1985. Rapid method to obtain bounds on accuracies and prediction error variances in mixed models. J. Dairy Sci. 68:2123.

ACKNOWLEDGMENTS

The author acknowledges 21st Century Genetics and Grant I-629-83-RE of the USA-Israel Binational Agricultural Development Project for financial support during this degree and thanks Dr. A. E. Freeman for again being a pleasure to work with. The author's officemates, Keith Boldman, Gwang-Joo Jeon, Martin Sieber, and Dusan Latinovic, all provided a most entertaining environment in which to work. Those who supplied data sets and those who served on my graduate committee also deserve thanks.

APPENDIX

Program ABSORBM

Fortran program to absorb herd-year-season equations and to output sire and group equations and other information needed by program REMLM. Program works for any number of traits provided they are all measured on every animal. Each trait is expected to behave according to a model of the form y = u + herd-year-season + genetic group + sire (within genetic group) + error.

To use this program:

1) Data must be sorted by herd-year-season and by sire within HYS and stored at the location described in FT10F001.
2) A list of the sire identification numbers which appear in the data (for unrelated sires) or which appear in the relationship matrix (for related sires) must be sorted in ascending order and stored in FT11F001.
3) If genetic groups are used, the genetic group to which each sire belongs should accompany the sire ID's in FT11F001. Group numbers must be consecutive integers starting with 1. If genetic groups are not used (NGM1 = 0) matrices having NGM1 as one of the dimensions should have this dimension set to 1 instead of zero or the program will not compile.
4) All parameters listed in the options segment of the program must be specified. These include the numbers of sires, traits, and genetic groups (minus 1), whether HYS should be treated as random

(should be used only when number of traits = 1) and if HYS are random, the ratio of error to HYS variance, and whether the user desires a listing of sire information.

5) Dimensions of all vectors and matrices must be altered according to values of NNS (number of sires), NT (number of traits), NGM1 (number of genetic groups minus 1), and NTCP (number of traits and cross-products) as follows:

```
REAL Z(NNS,NNS), RHS(NNS,NT), HOLD(NNS), Y(NT)
INTEGER LIST(NNS), LPROG(NNS), LHERD(NNS),
        IDLIST(NNS), NPLIST(NNS)
REAL*8 SSCP(NTCP), WSSB(NTCP), YSY(NTCP)
REAL*8 HTOT(NT), SUM(NT), RMU(NT), DMU, HYS, HYSL
REAL SXG(NNS,NGM1), GXG(NGM1,NGM1), GRHS(NGM1,NT)
INTEGER IGRP(NNS)
```

6) Format statements 4 and 13 must be specified for reading the sire list and the data.

7) Other fixed effects known to influence the data may be estimated ahead of time and subtracted from the data as correction factors as if these were known. This works well for fixed effects with few degrees of freedom.

8) Be sure that the traits included are not linear function of each other. This would cause a singular error variance-covariance matrix which the program is not prepared to deal with.

9) Group equations are assumed to be full rank after deleting the last row. This requires connectedness among all groups specified.

10) Make changes to JCL for time, region, and data set descriptions. Specifically, storage space for matrices output to FT12F001 will rise proportionally to number of sires squared.

Program REMLM

Fortran program to compute multiple-trait REML estimates of genetic and error variance and covariance components and to obtain estimates of sire transmitting abilities calculated as if estimated variance matrices were true parameters.

To use this program:

1) First run program ABSORBM which outputs matrices needed by this program.
2) Parameters listed in the options segment of the program must be specified. Numbers of sires, traits, and genetic groups (minus 1) must be the same as were used in ABSORBM. One exception is that NGM1 can be set to zero even though a positive number of group equations were created in ABSORBM.
3) If sires are related, the nonzero coefficients of A^{-1} should be stored in the location described in FT11F001 and the variable RELATD set equal to 1.
4) Dimensions of all vectors and matrices must be altered according to values of NNS, NT, NGM1, and NTCP as follows:

```
REAL AINV(NNS**2+3*NNS), WKAREA(NNS,NNS), WKB(NNS)
REAL Z(NNS,NNS), ZINV(NNS,NNS), ETA(NNS,NT), RHS(NNS,NT), HOLD(NNS)
REAL RHSA(NNS,NT), R(NT,NT), G(NT,NT), S(NT,NT), P(NT,NT)
REAL PINV(NT,NT), TV(NT,NT), TINV(NT,NT), STOR(NT,NT), WK(NT), D(NT)
REAL RV(NTCP), UPU(NTCP)
REAL*8 YSY(NTCP), WSSB(NTCP), SUM, COEF
REAL*8 TRU(NT), TRL(NT), DIAG
REAL SXG(NNS,NGM1), SXGB(NNS,NGM1), GXG(NGM1,NGM1), GXGINV(NGM1,NGM1)
REAL GRHS(NGM1,NT), GSOL(NGM1,NT), GRPN(NGM1)
INTEGER LIST(NNS), LISTB(NNS), IGRP(NNS)
```

5) The variable JTYPE determines whether an exact procedure using diagonalization of $Z'SZ$ and A^{-1} or an approximate procedure involving estimation of traces will be used. For small problems, JTYPE = 1 should always be specified. For certain larger problems, JTYPE = 0 can save money but this produces approximate REML estimates. JTYPE = 0 is probably cheaper if the number of traits times the expected number of iterations times 7 is less than the number of sires.

6) Make changes to JCL for time, region, and data set descriptions. Region required (in kilobytes) is approximately .015 times the number of sires squared.

Computational Procedures Used in ABSORBM and REMLM

1) Absorption

   a) Because all traits have equal information and equal models, only one sire coefficient matrix is set up even with many right-hand-sides.

   b) If HYS are fixed, HYS equations are absorbed into sire equations.

   c) If HYS are random, a ratio is added to HYS equations and these are absorbed into sire equations and μ equation. Then, the μ equation is absorbed into sire equations.

   d) Genetic group equations are constructed from the sire equations. The last genetic group is set equal to zero (deleted) and genetic group equations are absorbed into the sire equations in program REMLM.

2) Priors

   a) Prior estimates of error variances and covariances are obtained from the within smallest subclass sums of squares and cross-products divided by degrees of freedom. If this estimator has fewer than 10 degrees of freedom the within HYS sums of squares and cross-products are used instead.

   b) Prior estimates for genetic variances and covariances are obtained by dividing the prior estimates of error variances and covariances by an appropriate scalar determined by the user.

3) Diagonalization

   a) If JTYPE = 1 is specified, the sire coefficient matrix Z'SZ is diagonalized by eigenvector-eigenvalue analysis.

   b) If JTYPE = 1 and sires are related, a matrix L is chosen such that $L'^{-1}L^{-1} = A^{-1}$. Eigenvector-eigenvalue analysis is then performed on L'(Z'SZ)L, which simultaneously diagonalizes Z'SZ and A^{-1}.

   c) For multiple-trait problems, the genetic (G) and error (R) variance-covariance matrices are simultaneously diagonalized at each round of iteration.

4) Bounds

   a) If JTYPE = 0 is specified, traces of inverses of sire coefficient matrices are approximated by averages of upper and lower bounds.

b) If JTYPE = 0 and sires are related, trace of $A^{-1}(Z'SZ + A^{-1}k)^{-1}$ is approximated by inverting N by N segments of $Z'SZ + A^{-1}k$, where N varies and is the number of nonzero elements in any row of A^{-1}.

5) Convergence

a) Convergence of the REML algorithm is declared when all genetic variances are changing by no more than .001 times their current value and no genetic correlations are changing by more than .001 absolutely.

b) Convergence of the Gauss-Seidel algorithm for obtaining sire solutions is declared when the standard deviation of changes in solutions is less than .0005 times the standard deviation of solutions.

6) Standard errors

a) Standard errors of REML solutions are obtained by calculating the size of a balanced data set which would have the same trace of inverse as that in the unbalanced data. Then, standard errors are calculated as if an unbiased procedure was used on the balanced data.

b) Standard errors of heritabilities are from the method of Swiger et al. (1964).

c) Standard errors of group solutions are averages of upper and lower bounds. Group solutions themselves measure the difference of any group from the last group.

```
//ABSORBM JOB
//S1 EXEC FORTG
//FORT.SYSIN DD *
      REAL Z(10,10),RHS(10,2),HOLD(10),Y(2)
      INTEGER LIST(10),LPROG(10),LHERD(10),IDLIST(10),NPLIST(10)
      REAL*8 SSCP(3),WSSB(3),YSY(3)
      REAL*8 HTOT(2),SUM(2),RMU(2),DMU,HYS,HYSL
      REAL SXG(10,1),GXG(1,1),GRHS(1,2)
      INTEGER IGRP(10)
C
C                     OPTIONS FOR PROGRAM ABSORBM:
C                       NNS    IS NUMBER OF SIRES
C                       NT     IS NUMBER OF TRAITS
C                       NGM1   IS NUMBER OF GENETIC GROUPS MINUS 1
C                              =0 MEANS GENETIC GROUPS NOT USED
C                       RANDOM=0 MEANS THAT HYS ARE FIXED
C                              =1 MEANS THAT HYS ARE RANDOM
C                       LSTOUT=0 MEANS NO LISTING OF SIRE INFO.
C                              =1 MEANS PRINT SIRE ID'S AND PROG.NO.
C                       X      IS RATIO OF ERROR VARIANCE TO HYS VAR.
C                       LIST   IS LIST OF SORTED SIRE ID'S
      NNS=10
      NT=2
      NGM1=1
      RANDOM=0.
      LSTOUT=1
      X=8.16
      NTCP=NT*(NT+1)/2
      DO 3 I=1,NNS
        READ(11,4) LIST(I),IGRP(I)
3       IF(NGM1 .EQ. 0) IGRP(I)=1
4     FORMAT(I9,I5)
      DO 5 IT=1,NT
        RMU(IT)=0.D0
        SUM(IT)=0.D0
5       HTOT(IT)=0.D0
      DO 6 I=1,NTCP
        YSY(I)=0.D0
        SSCP(I)=0.D0
6       WSSB(I)=0.D0
      DO 9 I=1,NNS
        HOLD(I)=0.0
        LPROG(I)=0
        LHERD(I)=0
        DO 7 J=1,NNS
7         Z(I,J)=0.0
        DO 8 IT=1,NT
8         RHS(I,IT)=0.0
9       CONTINUE
      NNH=0
```

```
      HYSL=0.
      LSIRE=0
      NTOT=0
      MTOT=0
      NH=0
      NS=0
      NP=0
      DMU=0.D0
C                              BEGIN LOOP TO READ DATA
10    READ(10,13,END=14) IHRD,IDSIRE,Y(1),Y(2)
13    FORMAT(I4,I5,2F10.3)
      HYS=IHRD
      GO TO 15
14    HYS=0.D0
      IDSIRE=0
15    CONTINUE
C                              CHECK IF OBS IS FROM NEW SIRE,
C                           THEN CALCULATE WITHIN SMALLEST SUBCLASS STUFF.
      IF(HYS .NE. HYSL) GO TO 16
      IF(IDSIRE .EQ. LSIRE) GO TO 19
16      CONTINUE
        IF(LSIRE .EQ. 0) GO TO 19
        IND2=IFIND(LSIRE,LIST,NNS)
        IND=0
        DO 18 IT=1,NT
          DO 17 JT=1,IT
            IND=IND + 1
            WSSB(IND)=WSSB(IND)+SSCP(IND)-SUM(IT)*SUM(JT)/NP
            YSY(IND)=YSY(IND) + SSCP(IND)
            SSCP(IND)=0.0
17          CONTINUE
          RHS(IND2,IT)=RHS(IND2,IT) + SUM(IT)
18        HTOT(IT)=HTOT(IT) + SUM(IT)
        DO 185 IT=1,NT
185       SUM(IT)=0.D0
        NS=NS + 1
        IDLIST(NS)=IND2
        NPLIST(NS)=NP
        LPROG(IND2)=LPROG(IND2) + NP
        LHERD(IND2)=LHERD(IND2) + 1
        NH=NH + NP
        NP=0
19      CONTINUE
        LSIRE=IDSIRE
C                            CHECK IF OBSERVATION IS FROM NEW HERD.
C                            THEN, ABSORB EQUATION FROM LAST HERD.
      IF(HYS .EQ. HYSL) GO TO 26
        IF(NH .EQ. 0) GO TO 25
        IF(RANDOM .EQ. 0.) HN=NH
        IF(RANDOM .EQ. 1.) HN=NH + X
```

```
      DO 22 I=1,NS
        IND=IDLIST(I)
        Z(IND,IND)=Z(IND,IND) + NPLIST(I)
        HOLD(IND)=HOLD(IND) + NPLIST(I)*X/HN
        DO 20 J=1,NS
          IND2=IDLIST(J)
20        Z(IND,IND2)=Z(IND,IND2) - NPLIST(I)*NPLIST(J)/HN
        DO 21 IT=1,NT
21        RHS(IND,IT)=RHS(IND,IT) - HTOT(IT)*NPLIST(I)/HN
22      CONTINUE
      IND=0
      DO 23 IT=1,NT
        DO 23 JT=1,IT
          IND=IND + 1
23        YSY(IND)=YSY(IND) - HTOT(IT)*HTOT(JT)/NH
      DO 24 IT=1,NT
        RMU(IT)=RMU(IT) + HTOT(IT)*X/HN
24      HTOT(IT)=0.0
      DMU=DMU + NH - NH*NH/HN
      NNH=NNH + 1
      NTOT=NTOT + NH
      MTOT=MTOT + NS
      NH=0.0
      NS=0
25    CONTINUE
      HYSL=HYS
26    CONTINUE
C                              CHECK FOR END OF DATA (HYS=0)
      IF(HYS .EQ. 0.) GO TO 110
      NP=NP + 1
      IND=0
      DO 40 IT=1,NT
        DO 35 JT=1,IT
          IND=IND + 1
35        SSCP(IND)=SSCP(IND) + Y(IT)*Y(JT)
40      SUM(IT)=SUM(IT) + Y(IT)
C                              END LOOP FOR READING DATA
      GO TO 10
110   CONTINUE
      IND=0
      DO 120 IT=1,NT
        DO 115 JT=1,IT
          IND=IND + 1
          IF(NTOT-MTOT .LT. 10) WSSB(IND)=YSY(IND)/(NTOT-NNH)
115       IF(NTOT-MTOT .GE. 10) WSSB(IND)=WSSB(IND)/(NTOT-MTOT)
120     CONTINUE
C                              ABSORB MU EQUATION IF HYS RANDOM
      IF(RANDOM .EQ. 0.) GO TO 124
      DO 123 I=1,NNS
        DO 121 J=1,NNS
```

```
121       Z(I,J)=Z(I,J) - HOLD(I)*HOLD(J)/DMU
        DO 122 IT=1,NT
122       RHS(I,IT)=RHS(I,IT) - RMU(IT)*HOLD(I)/DMU
123     CONTINUE
124   CONTINUE
      WRITE(12) RHS
      WRITE(12) WSSB,YSY,NNH,NTOT,MTOT
      DO 126 I=1,NNS
        DO 125 J=1,NNS
125       HOLD(J)=Z(J,I)
126     WRITE(12) HOLD
      WRITE(12) LIST,IGRP
C                                 FORM AND WRITE OUT GROUP EQUATIONS
      IF(NGM1 .EQ. 0) GO TO 139
        DO 129 I=1,NGM1
          DO 127 J=1,NNS
127         SXG(J,I)=0.0
          DO 128 J=1,NGM1
128         GXG(I,J)=0.0
          DO 129 IT=1,NT
129         GRHS(I,IT)=0.0
        DO 132 I=1,NNS
          IND=IGRP(I)
          IF(IND .GT. NGM1) GO TO 132
          DO 131 J=1,NNS
131         SXG(J,IND)=SXG(J,IND) + Z(I,J)
132       CONTINUE
        GRPNL=0.0
        DO 135 J=1,NNS
          IND=IGRP(J)
          IF(IND .GT. NGM1) GO TO 135
          DO 133 I=1,NGM1
            GXG(IND,I)=GXG(IND,I) + SXG(J,I)
133         GRPNL=GRPNL + SXG(J,I)
          DO 134 IT=1,NT
134         GRHS(IND,IT)=GRHS(IND,IT) + RHS(J,IT)
135       CONTINUE
        WRITE(12) SXG
        WRITE(12) GXG,GRHS,GRPNL
139     CONTINUE
       PRINT 140,NTOT,MTOT,NNH,NNS
140    FORMAT('ONO. OF RECORDS,FILLED SUBCLASSES,HERDS,AND SIRES',
     */,' ',4I12)
       IF(LSTOUT .EQ. 0) GO TO 150
         PRINT 141
141      FORMAT('0        N     SIRE ID     GROUP     #PROG   #HERDS',
     *         '    EFF. NUM.')
         DO 142 I=1,NNS
142        PRINT 145,I,LIST(I),IGRP(I),LPROG(I),LHERD(I),Z(I,I)
145        FORMAT(' ',5I10,F12.2)
```

```
150     CONTINUE
      STOP
      END
      FUNCTION IFIND(ID,LIST,N)
      INTEGER LIST(N)
      I=1
      J=N
5     IF(I .LE. J) GO TO 20
        GO TO 70
20    K=(I+J)/2
      IF(ID .GT. LIST(K)) GO TO 60
      IF(ID .EQ. LIST(K)) GO TO 80
        J=K - 1
        GO TO 5
60    I=K + 1
      GO TO 5
70    PRINT 71,ID
71    FORMAT(' SIREID WAS NOT FOUND',I10,/,'0OBSERVATION WAS ',
     *       'ATTRIBUTED TO FIRST SIRE')
      IFIND=1
      RETURN
80    IFIND=K
      RETURN
      END
//GO.FT10F001 DD *
   1    1    21.452    99.252
   1    1    21.193   102.940
   1    1    20.102    97.053
   1    1    15.610   101.115
   1    4    21.753    99.498
   1    4    17.635    98.151
   1    4    17.301    98.208
   1    5    18.115   103.320
   1    5    19.691    99.040
   1    5    18.096    96.624
   1    7    21.695    99.011
   1    7    19.747   101.699
   1    7    18.190    99.839
   1    7    18.640   100.475
   1    8    21.863   100.338
   1    8    20.278   102.966
   1    8    19.036    95.209
   1    9    22.753   107.425
   2    6    20.535    97.043
   2    6    22.340   108.602
   2    6    21.858   100.212
   2    9    20.744    96.114
   2    9    25.811    98.612
   2    9    20.141   100.857
   3    2    19.942   100.198
```

3	2	17.885	98.922
3	3	20.602	101.337
3	4	22.583	98.136
3	4	20.959	98.389
3	5	13.799	95.213
3	7	20.633	101.923
3	7	19.461	96.552
3	8	20.785	100.059
3	8	20.173	96.874
3	8	20.248	100.059
3	8	19.835	100.559
4	8	18.479	99.616
4	9	21.499	102.735
4	9	20.597	105.478
4	10	23.001	98.764
4	10	21.717	98.226
5	3	17.992	97.635
5	3	18.390	105.155
5	6	17.793	102.643
5	6	19.624	96.256
5	6	21.908	100.518
5	6	20.833	98.929
5	8	21.593	99.286
5	9	20.695	101.417
5	9	22.094	96.795
5	9	21.574	101.871
5	9	22.054	100.140
6	2	22.435	98.995
6	2	21.791	97.937
6	2	17.319	96.914
6	5	17.613	105.889
6	5	17.380	99.893
6	6	17.250	106.584
6	6	19.635	101.783
6	8	15.466	99.050
6	8	20.455	100.039
6	8	19.435	104.977
6	10	22.648	95.979
6	10	26.222	96.139
7	3	26.296	100.342
7	6	20.806	97.503
7	6	20.763	98.159
7	10	22.614	98.861
7	10	23.448	102.157
8	3	18.982	98.954
8	3	20.153	96.993
8	3	21.037	103.463
8	3	19.591	98.970
8	4	21.092	93.855
8	4	22.272	97.457

```
  8     4    21.337    97.699
  8     4    20.634   101.801
  8     5    20.079   100.248
  8     5    16.552   100.185
  8     5    18.584   105.355
  8     5    15.844    98.968
  8     8    23.231   100.079
  8     8    17.003   102.937
  8     9    19.926   100.963
  9     5    19.556    94.021
  9     5    18.436   101.534
  9     5    20.937    98.286
  9     5    22.219    94.895
  9     6    18.650   101.436
  9     7    22.398   103.256
  9     7    21.380    99.087
  9     7    21.741   102.051
  9     8    19.918   102.196
  9     8    18.795    97.540
  9     8    16.233   101.117
  9     8    18.818   102.566
  9     9    20.229    99.627
  9    10    21.042   100.282
 10     2    22.067    93.227
 10     2    21.902    92.366
 10     5    17.589   101.728
 10     7    20.218    98.640
 10     8    21.940    96.117
 10     8    19.244    99.651
 10     8    18.784   100.362
 10     9    19.100   102.945
 10     9    22.482   105.683
//GO.FT11F001 DD *
       1     1
       2     1
       3     1
       4     1
       5     1
       6     2
       7     2
       8     2
       9     2
      10     2
//GO.FT12F001 DD UNIT=DISK,DISP=(NEW,CATLG),DSN=P.I6280.PASS,
//   SPACE=(TRK,(2,2),RLSE),DCB=(RECFM=VS)
```

```
//REMLM JOB
//S1 EXEC FORTG,REGION.GO=1024K
//FORT.SYSIN DD *
      REAL AINV(130),WKAREA(10,10),WKB(10)
      EQUIVALENCE (AINV,WKAREA)
      REAL Z(10,10),ZINV(10,10),ETA(10,2),RHS(10,2),HOLD(10)
      REAL RHSA(10,2),R(2,2),G(2,2),S(2,2),P(2,2)
      REAL PINV(2,2),TV(2,2),TINV(2,2),STOR(2,2),WK(2),D(2)
      REAL RV(3),UPU(3)
      REAL*8 YSY(3),WSSB(3),SUM,COEF
      REAL*8 TRU(2),TRL(2),DIAG
      REAL SXG(10,1),SXGB(10,1),GXG(1,1),GXGINV(1,1)
      REAL GRHS(1,2),GSOL(1,2),GRPN(1)
      INTEGER LIST(10),LISTB(10),IGRP(10)
C
C                          OPTIONS FOR PROGRAM REMLM:
C                            NNS IS THE NUMBER OF SIRES
C                            NT  IS THE NUMBER OF TRAITS
C                            SINGLE=0 MEANS MULTIPLE-TRAIT ANALYSIS
C                                  =1 MEANS SINGLE-TRAIT ANALYSIS
C                            JTYPE =0 MEANS THAT BOUNDS WILL BE USED
C                                  =1 MEANS DIAGONALIZATION OF ZSZ
C                            RELATD=0 IS FOR SIRES UNRELATED
C                                  =1 MEANS THAT AINV MUST BE SUPPLIED
C                            NGM1  =NUMBER OF GENETIC GROUPS - 1
C                                  =0 MEANS GENETIC GROUPS NOT USED
C                            ETAOUT=0 MEANS NO OUTPUT OF ETA'S
C                                  =1 MEANS OUTPUT ON PAPER
C                                  =2 MEANS OUTPUT TO FT12F001
C                                  =3 MEANS OUTPUT TO BOTH
C                            X   IS THE PRIOR ESTIMATE SIGMA0/SIGMA1,
C                                OR AVE. OF RATIOS FOR MULTI-TRAIT
C                            IQUIT IS MAXIMUM ITERATIONS ALLOWED
C                            NTCP   IS NUMBER OF TRAITS + CROSSPROD.
C                            DIMENSION OF AINV IS NNS**2 + 3*NNS
      NNS=10
      NT=2
      SINGLE=0.
      JTYPE=1
      RELATD=1.
      NGM1=1
      ETAOUT=1.
      X=9.
      IQUIT=30
      NTCP=NT*(NT+1)/2
      NS2=NNS*(NNS+1)/2
      NG=NGM1 + 1
      READ(10) RHS
      READ(10) WSSB,YSY,NNH,NTOT,MTOT
      DO 5 I=1,NNS
```

```
      DO 4 IT=1,NT
        ETA(I,IT)=0.0
4       RHSA(I,IT)=RHS(I,IT)
5     CONTINUE
      PRINT 12
12    FORMAT('OWSSUB ESTIMATES OF ERROR VARIANCES')
      DO 15 IT=1,NT
        DO 13 JT=1,IT
          IND=IT*(IT-1)/2 + JT
          RV(IND)=WSSB(IND)
          R(IT,JT)=RV(IND)
          R(JT,IT)=RV(IND)
          G(IT,JT)=RV(IND)/X
13        G(JT,IT)=G(IT,JT)
        PRINT 14,IT,R(IT,IT)
14      FORMAT(' ',I5,3F14.6)
        WK(IT)=0.0
15      CONTINUE
C                          READ IN ZSZ AND AINV (IF PRESENT),
C                          PERFORM SIMULTANEOUS DIAGONALIZATION OF
C                          ZSZ AND AINV OR DIAGONALIZATION OF ZSZ IF
C                          SIRES ARE UNRELATED.
      IF(RELATD .EQ. 0.) GO TO 24
      DO 16 J=1,NS2
16      AINV(J)=0.0
17    READ(11,18,END=19) IROW,ICOL,COEF
18    FORMAT(2I5,D10.4)
      IF(ICOL .GT. IROW) GO TO 17
      IF(IROW .GT. NNS) GO TO 17
      IND=IROW*(IROW-1)/2 + ICOL
      AINV(IND)=COEF
      GO TO 17
19    IJOB=2
      IF(JTYPE .EQ. 0) GO TO 24
      CALL LINV3P(AINV,HOLD,IJOB,NNS,IER)
      IND=0
      DO 21 J=1,NNS
        DO 20 K=1,J
          IND=IND + 1
          Z(K,J)=0.0
20        Z(J,K)=AINV(IND)
21      Z(J,J)=1./Z(J,J)
      IDGT=4
      CALL LINV2F(Z,NNS,NNS,ZINV,IDGT,AINV,IER)
C                            READ IN ABSORBED COEFFICIENT MATRIX(ZSZ)
24    DO 25 J=1,NNS
        READ(10) WKB
        DO 25 K=1,NNS
25        Z(K,J)=WKB(K)
      READ(10) LIST,IGRP
```

```
      IF(NGMI .EQ. 0) GO TO 29
C                              READ IN AND ABSORB GROUP EQUATIONS
C                              ADJUST YSY, SUBTRACT GRHS'GXGINV GRHS
        READ(10) SXG
        READ(10) GXG,GRHS,GRPNL
        DO 26 J=1,NGM1
26        GRPN(J)=GXG(J,J)
        IDGT=4
        CALL LINV1F(GXG,NGM1,NGM1,GXGINV,IDGT,WKB,IER)
        CALL MULT(NNS,NGM1,NGM1,SXGB,SXG,GXGINV,0,0)
        CALL MULT(NGM1,NGM1,NT,GSOL,GXGINV,GRHS,0,0)
        DO 926 J=1,NT
          DO 926 K=1,J
            IND=J*(J-1)/2 + K
            DO 926 L=1,NGM1
926           YSY(IND)=YSY(IND) - GRHS(L,J)*GSOL(L,K)
        DO 28 J=1,NNS
          DO 927 K=1,NNS
            DO 927 L=1,NGM1
927           Z(J,K)=Z(J,K) - SXGB(J,L)*SXG(K,L)
          DO 28 IT=1,NT
            SUM=0.D0
            DO 27 K=1,NGM1
27            SUM=SUM + SXGB(J,K)*GRHS(K,IT)
            RHSA(J,IT)=RHSA(J,IT) - SUM
28          RHS(J,IT)=RHSA(J,IT)
29      CONTINUE
      IF(JTYPE .EQ. 0) GO TO 33
      IF(RELATD .EQ. 0.) GO TO 30
C                              ADJUST RHS BY L (ZINV) MATRIX
C                              MULTIPLY TO GET L(ZSZ)L'
      CALL MULT(NNS,NNS,NT,RHSA,ZINV,RHS,0,0)
      CALL MULT(NNS,NNS,NNS,WKAREA,Z,ZINV,0,1)
      CALL MULT(NNS,NNS,NNS,Z,ZINV,WKAREA,0,0)
30    JOBN=12
      CALL EIGRS(Z,NNS,JOBN,HOLD,ZINV,NNS,AINV,IER)
      CALL MULT(NNS,NNS,NT,RHS,ZINV,RHSA,1,0)
33      CONTINUE
      BDIFF=0.
C                             LOOP TO DO ITERATION OF REML PROCEDURE
      DO 100 I=1,IQUIT
        IF(SINGLE .EQ. 0.) GO TO 40
          DO 38 J=1,NNS
            DO 38 IT=1,NT
38            RHSA(J,IT)=RHS(J,IT)
          GO TO 60
40      IND=0
        DO 45 J=1,NT
         DO 45 K=1,J
          IND=IND + 1
```

```
45        RV(IND)=R(J,K)
C                    DECOMPOSITION RV=TV*TV'   WHERE TV IS
C                    A LOWER TRIANGULAR MATRIX
       IJOB=2
       CALL LINV3P(RV,WK,IJOB,NT,IER)
         IND=0
         DO 47 J=1,NT
           DO 46 K=1,J
             IND=IND + 1
             TV(K,J)=0.0
46           TV(J,K)=RV(IND)
47         TV(J,J)=1./TV(J,J)
C                    MULTIPLY TO GET  TINV*G*TINV'
       IDGT=4
       CALL LINV2F(TV,NT,NT,TINV,IDGT,WKB,IER)
      CALL MULT(NT,NT,NT,S,G,TINV,0,1)
      CALL MULT(NT,NT,NT,STOR,TINV,S,0,0)
C                        GET EIGENVALUES OF STOR
      JOBN=12
      CALL EIGRS(STOR,NT,JOBN,D,S,NT,WKB,IER)
C                      FORM THE PRODUCT  S'*TINV = PINV
C                           SAVE COPY OF PINV IN TV
        CALL MULT(NT,NT,NT,PINV,S,TINV,1,0)
        DO 58 J=1,NT
         DO 58 K=1,NT
58         TV(J,K)=PINV(J,K)
C                             ADJUST RIGHT-HAND SIDES
        CALL MULT(NNS,NT,NT,RHSA,RHS,PINV,0,1)
C                       FIND INVERSE OF EIGENVALUES
C                       AND APPLY APPROPRIATE CONSTRAINTS
60      DO 965 J=1,NT
          IF(SINGLE .EQ. 1.) D(J)=G(J,J)/R(J,J)
          TRU(J)=0.D0
          TRL(J)=0.D0
          IF(D(J) .GT. .33333) D(J)=.33333
          IF(D(J) .LT. 1.0E-5) D(J)=1.0E-5
965       D(J)=1./D(J)
        PRINT 966,D
966     FORMAT('0 RATIOS FOR DIAGONAL',8F12.6,/,(21X,8F12.6,/))
C                       SOLVE FOR ETA AND GET TRACES FOR
C                       DIAGONALIZED VERSION OF ZSZ
        IF(JTYPE .EQ. 0) GO TO 62
        DO 969 J=1,NNS
          DIAG=HOLD(J)
          DO 968 IT=1,NT
            ETA(J,IT)=RHSA(J,IT)/(DIAG+D(IT))
968         TRL(IT)=TRL(IT) + 1.D0/(DIAG+D(IT))
969       CONTINUE
        GO TO 63
62      CALL SOLVE(Z,AINV,D,RHSA,ETA,UPU,NNS,NT,NTCP,NS2,RELATD)
```

```
C
C                    GET U'Z'MY FOR REML ERROR QUADRATIC
C                    STORE THIS IN TINV
63      DO 971 J=1,NT
          DO 971 K=1,NT
            SUM=0.DO
            DO 970 L=1,NNS
970           SUM=SUM + ETA(L,J)*RHSA(L,K)
971         TINV(J,K)=SUM
C                    FORM CROSS-PRODUCTS OF SOLUTIONS U'U
C                    OR U'AINV U IF ANIMALS ARE RELATED
          IF(JTYPE .EQ. 1 .OR. RELATD .EQ. 0.) GO TO 66
          DO 65 K=1,NNS
            DO 972 IT=1,NT
972           RHSA(K,IT)=0.0
            DO 65 L=1,NNS
              IF(L .LE. K) IND=K*(K-1)/2 + L
              IF(L .GT. K) IND=L*(L-1)/2 + K
              IF(AINV(IND) .EQ. 0.) GO TO 65
              DO 64 IT=1,NT
64              RHSA(K,IT)=RHSA(K,IT) + AINV(IND)*ETA(L,IT)
65            CONTINUE
          GO TO 68
66        DO 67 IT=1,NT
            DO 67 K=1,NNS
67            RHSA(K,IT)=ETA(K,IT)
68        DO 70 J=1,NT
            DO 70 K=1,J
              IND=J*(J-1)/2 + K
              UPU(IND)=0.0
              DO 70 L=1,NNS
70              UPU(IND)=UPU(IND) + RHSA(L,J)*ETA(L,K)
C                    COMPUTE BOUNDS ON TRACE IF JTYPE = 0
      IF(JTYPE .EQ. 1) GO TO 74
      IF(RELATD .EQ. 1.) GO TO 72
      CALL BOUNDS(Z,TRU,TRL,D,NNS,NT)
      DO 71 IT=1,NT
        X1=NNS - TRL(IT)*D(IT)
        X2=NNS - TRU(IT)*D(IT)
        X1=(X1 - X2)*2./(X1 + X2)
        IF(X1 .GT. BDIFF) BDIFF=X1
71      TRL(IT)=.5*(TRU(IT)+TRL(IT))
      GO TO 74
72    CALL BOUNDA(Z,AINV,HOLD,LISTB,WKB,TRU,TRL,D,NNS,NT,NS2)
74      CONTINUE
C                    COMPUTE VARIANCES OF TRANSFORMED TRAITS
      DO 76 J=1,NT
       DO 75 K=1,J
         S(J,K)=G(J,K)
         IND=J*(J-1)/2 + K
```

```
          DENOM=DSQRT((NNS - TRL(J)*D(J))*(NNS - TRL(K)*D(K)))
          G(J,K)=UPU(IND)/DENOM
75        G(K,J)=UPU(IND)/DENOM
76        CONTINUE
        DO 80 J=1,NT
          IF(SINGLE .EQ. 0.) GO TO 78
            IF(G(J,J)/R(J,J) .GT. .33333) G(J,J)=.33333*R(J,J)
            GO TO 80
78        IF(G(J,J) .LT. .33333) GO TO 80
          RATIO=SQRT(.33333/G(J,J))
          DO 79 K=1,NT
            G(J,K)=G(J,K)*RATIO
79          G(K,J)=G(K,J)*RATIO
80        CONTINUE
        IF(SINGLE .EQ. 1.) GO TO 82
C                       ROTATE BACK TO ORIGINAL TRAITS
        IDGT=2
        CALL LINV1F(PINV,NT,NT,P,IDGT,WK,IER)
         CALL MULT(NT,NT,NT,STOR,G,P,0,1)
         CALL MULT(NT,NT,NT,G,P,STOR,0,0)
          CALL MULT(NT,NT,NT,STOR,TINV,P,0,1)
          CALL MULT(NT,NT,NT,TINV,P,STOR,0,0)
C
C                           CHECK FOR CONVERGENCE
82        CONV=1
          DO 85 J=1,NT
           DO 85 K=1,J
             DIFF=(G(J,K)-S(J,K))/SQRT(S(J,J)*S(K,K))
85           IF(ABS(DIFF) .GT. .001) CONV=0.0
C                           GET NEW ESTIMATE OF ERROR VARIANCE
           DO 90 J=1,NT
             DO 88 K=1,NT
               IF(K .LE. J) IND=J*(J-1)/2 + K
               IF(K .GT. J) IND=K*(K-1)/2 + J
88             R(J,K)=(YSY(IND) - TINV(J,K))/(NTOT - NNH - NGM1)
90           CONTINUE
         PRINT 99,I
99       FORMAT('0ROUND',I4,' SUCCESSFULLY COMPLETED.')
         IF(CONV .EQ. 1.) GO TO 110
100      CONTINUE
        PRINT 105,IQUIT
105     FORMAT('0CONVERGENCE CRITERION NOT MET AFTER',I5,'  ROUNDS')
110     CONTINUE
        IF(D(1) .GE. 9.9E4) PRINT 113
113     FORMAT('0GENETIC VARIANCE MATRIX APPROACHING SINGULARITY')
        IF(JTYPE .EQ. 1) GO TO 117
          IF(RELATD .EQ. 0.) PRINT 114,BDIFF
114       FORMAT('0LARGEST DIFFERENCE BETWEEN BOUNDS WAS',
       *        F8.5,'  TIMES THE AVERAGE OF BOUNDS')
          IF(RELATD .EQ. 1.) PRINT 115
```

```
115     FORMAT('0WARNING: APPROXIMATE TRACES WERE USED')
117   PRINT 118
118   FORMAT('0FINAL ESTIMATES OF ERR VAR.,SIRE VAR, AND HERITABILITY')
      DO 120 IT=1,NT
        H2=4.*G(IT,IT)/(G(IT,IT) + R(IT,IT))
        WK(IT)=R(IT,IT)/G(IT,IT)
120     PRINT 14,IT,R(IT,IT),G(IT,IT),H2
      PRINT 125
125   FORMAT('0STANDARD ERRORS OF CORRESPONDING PARAMETERS ABOVE')
      IF(JTYPE .EQ. 1) GO TO 127
        IF(RELATD .EQ. 0.) CALL BOUNDS(Z,TRU,TRL,WK,NNS,NT)
        IF(RELATD .EQ. 1.) CALL BOUNDA(Z,AINV,HOLD,LISTB,WKB,
     *                                 TRL,TRU,WK,NNS,NT,NS2)
127   CONTINUE
      DO 130 IT=1,NT
        XK=R(IT,IT)/G(IT,IT)
        VE=R(IT,IT)*SQRT(2./(NTOT - MTOT))
        TVAR=G(IT,IT) + R(IT,IT)
        H2=4.*G(IT,IT)/TVAR
        IF(JTYPE .EQ. 0) GO TO 129
          SUM=0.D0
          DO 128 J=1,NNS
128         SUM=SUM + 1./(HOLD(J) + XK)
          TRU(IT)=SUM
129     CONTINUE
        EFFN=XK/(NNS/(NNS-TRU(IT)*XK) - NNS/(NNS-1.))
        VG=2.*(NNS-1.+(NNS-1.)**2/(NTOT-MTOT))*(R(IT,IT)/EFFN/NNS)**2
        VG=VG + 4.*R(IT,IT)*G(IT,IT)/EFFN/NNS
        VG=VG + 2.*G(IT,IT)**2/(NNS-1.)
        VH=VG/G(IT,IT)**2 + (VE**2 + VG)/TVAR**2
        VH=SQRT(VH - 2.*VG/G(IT,IT)/TVAR)*H2
        VG=SQRT(VG)
        PRINT 14,IT,VE,VG,VH
130     CONTINUE
      IF(NT .EQ. 1) GO TO 150
      IF(SINGLE .EQ. 1.) GO TO 150
        PRINT 140
140     FORMAT('0    TRAITS       PHENOTYPIC COV. AND CORR.       ',
     *    '    ERROR COV. AND CORR.         GENETIC COV. AND CORR.')
        DO 145 IT=2,NT
          ITM1=IT - 1
          DO 145 JT=1,ITM1
            ECORR=R(IT,JT)/SQRT(R(IT,IT)*R(JT,JT))
            GCORR=G(IT,JT)/SQRT(G(IT,IT)*G(JT,JT))
            PCOV=G(IT,JT) + R(IT,JT)
            PCORR=PCOV/SQRT((G(IT,IT)+R(IT,IT))*(G(JT,JT)+R(JT,JT)))
            PRINT 143,IT,JT,PCOV,PCORR,R(IT,JT),ECORR,G(IT,JT),GCORR
143         FORMAT(' ',2I5,6F15.5)
145         CONTINUE
150   IF(ETAOUT .EQ. 0.) GO TO 200
```

```
      IF(JTYPE .EQ. 0) GO TO 180
        REWIND 10
        READ(10) RHS
        READ(10) WSSB,YSY,NNH,NTOT,MTOT
        DO 151 J=1,NNS
          READ(10) WKB
          DO 151 K=1,NNS
151         Z(K,J)=WKB(K)
        IF(NGM1 .EQ. 0) GO TO 158
          DO 154 J=1,NNS
            DO 153 K=1,NNS
              DO 153 L=1,NGM1
153             Z(J,K)=Z(J,K) - SXGB(J,L)*SXG(K,L)
            DO 154 IT=1,NT
              DO 154 K=1,NGM1
154             RHS(J,IT)=RHS(J,IT) - SXGB(J,K)*GRHS(K,IT)
158       CONTINUE
        IF(SINGLE .EQ. 0.) CALL MULT(NNS,NT,NT,RHSA,RHS,TV,0,1)
        DO 160 J=1,NNS
          DO 160 K=1,NT
            IF(SINGLE .EQ. 1.) RHSA(J,K)=RHS(J,K)
160         ETA(J,K)=RHSA(J,K)/(Z(J,J) + D(K))
        IF(RELATD .EQ. 0.) GO TO 165
          REWIND 11
          DO 162 J=1,NS2
162         AINV(J)=0.0
163       READ(11,164,END=165) IROW,ICOL,COEF
164       FORMAT(2I5,D10.4)
          IF(ICOL .GT. IROW) GO TO 163
          IF(IROW .GT. NNS) GO TO 163
          IND=IROW*(IROW-1)/2 + ICOL
          AINV(IND)=COEF
          GO TO 163
165     CALL SOLVE(Z,AINV,D,RHSA,ETA,UPU,NNS,NT,NTCP,NS2,RELATD)
180   IF(SINGLE .EQ. 0.) CALL MULT(NNS,NT,NT,RHSA,ETA,P,0,1)
      IF(SINGLE .EQ. 0.) GO TO 182
        DO 181 J=1,NNS
          DO 181 IT=1,NT
181         RHSA(J,IT)=ETA(J,IT)
182   IF(NGM1 .EQ. 0) GO TO 192
C                           SOLVE FOR GROUP EFFECTS
        DO 184 J=1,NGM1
          DO 184 IT=1,NT
            SUM=0.D0
            DO 183 K=1,NNS
183           SUM=SUM + SXG(K,J)*RHSA(K,IT)
184         GRHS(J,IT)=GRHS(J,IT) - SUM
        CALL MULT(NGM1,NGM1,NT,GSOL,GXGINV,GRHS,0,0)
C                           ADD GROUP SOL'NS TO SIRE SOL'NS
        DO 185 J=1,NNS
```

```
          DO 185 IT=1,NT
            IND=IGRP(J)
            IF(IND .GT. NGM1) GO TO 185
              RHSA(J,IT)=RHSA(J,IT) + GSOL(IND,IT)
185         CONTINUE
        PRINT 186
186     FORMAT('0',12X,'GROUP  EFF.NUM.     GROUP SOLUTIONS')
        DO 187 J=1,NGM1
187       PRINT 195,J,J,GRPN(J),(GSOL(J,IT),IT=1,NT)
        ZERO=0.0
        PRINT 195,NG,NG,GRPNL,(ZERO,IT=1,NT)
        PRINT 188
188     FORMAT('0                STANDARD ERRORS OF GROUP SOLUTIONS')
        DO 191 J=1,NGM1
          SUM=0.D0
          DO 189 K=1,NNS
189         SUM=SUM + SXGB(K,J)**2
          DO 190 IT=1,NT
190         WK(IT)=DSQRT((GXGINV(J,J) + SUM/D(IT)/2.)*R(IT,IT))
191       PRINT 195,J,J,GRPN(J),(WK(IT),IT=1,NT)
        PRINT 195,NG,NG,GRPNL,(ZERO,IT=1,NT)
192   IF(ETAOUT .NE. 2.) PRINT 193
193   FORMAT('0  SIRE ID  GRP   EFF.NUM.           ESTIMATED ',
     *       'TRANSMITTING ABILITIES')
      DO 197 J=1,NNS
        IF(ETAOUT .NE. 2.) PRINT 195,LIST(J),IGRP(J),Z(J,J),
     *                                (RHSA(J,K),K=1,NT)
195     FORMAT(' ',I9,I5,F12.2,6X,8F12.6,/,(33X,8F12.6,/))
        IF(ETAOUT .GT. 1.) WRITE(12,196) LIST(J),IGRP(J),Z(J,J),
     *                                   (RHSA(J,K),K=1,NT)
196     FORMAT(I9,I5,(F12.6))
197     CONTINUE
200   CONTINUE
      STOP
      END
      SUBROUTINE SOLVE(Z,AINV,XV,RHS,ETA,UPU,NNS,NT,NTCP,NS2,RELATD)
      REAL Z(NNS,NNS),AINV(NS2),RHS(NNS,NT),ETA(NNS,NT),UPU(NTCP),XV(NT)
      REAL*8 X,DMAX,DSUM,TOT,SUM
      IND=0
C                               LOOP TO DO NUMBER OF TRAITS
      DO 30 K=1,NT
        IND=IND + K
        X=XV(K)
C                               FORM  ZSZ+AINV*K   OR   ZSZ+IK
        IF(RELATD .EQ. 1.) GO TO 4
        DO 3 I=1,NNS
          AINV(I)=Z(I,I)
3       Z(I,I)=Z(I,I) + X
        GO TO 7
4       IND2=0
```

```
      DO 6 I=1,NNS
        DO 5 J=1,I
          IND2=IND2 + 1
          IF(AINV(IND2) .EQ. 0.) GO TO 5
          TEMP=Z(I,J)
          Z(I,J)=Z(I,J) + AINV(IND2)*X
          Z(J,I)=Z(I,J)
          IF(TEMP .EQ. 0.) TEMP=-99999.
          AINV(IND2)=TEMP
5         CONTINUE
6       CONTINUE
C                                 LOOP TO ITERATE FOR SIRE SOLUTIONS
7     DO 20 ITS=1,20
        DMAX=0.0
        DSUM=0.D0
        TOT=0.0
        UPU(IND)=0.0
        DO 10 I=1,NNS
          OLD=ETA(I,K)
          SUM=RHS(I,K) + Z(I,I)*ETA(I,K)
          DO 8 J=1,NNS
8           SUM=SUM - Z(I,J)*ETA(J,K)
          ETA(I,K)=SUM/Z(I,I)
          UPU(IND)=UPU(IND) + ETA(I,K)**2
          TOT=TOT + ETA(I,K)
          DIFF=ABS(ETA(I,K)-OLD)
          IF(DIFF .GT. DMAX) DMAX=DIFF
10        DSUM=DSUM + DIFF**2
          IF(DSUM/UPU(IND) .LT. .0005**2) GO TO 21
          ADD=TOT/NNS
          IF(RELATD .EQ. 1) GO TO 20
          DO 18 I=1,NNS
18          ETA(I,K)=ETA(I,K) - ADD
20        CONTINUE
21      CONTINUE
         IF(NT .EQ. 1) PRINT 22,ITS,DMAX
22       FORMAT('  CONVERGENCE AT',I4,' LARGEST CHANGE=',F10.6)
C
C                                 RESTORE ZSZ AND AINV TO ORIGINAL FORM
        IF(RELATD .EQ. 1.) GO TO 24
        DO 23 I=1,NNS
23        Z(I,I)=AINV(I)
        GO TO 27
24      IND2=0
        DO 25 I=1,NNS
          DO 25 J=1,I
            IND2=IND2 + 1
            IF(AINV(IND2) .EQ. 0.) GO TO 25
            TEMP=AINV(IND2)
            IF(TEMP .EQ. -99999.) TEMP=0.
```

```
            AINV(IND2)=(Z(I,J) - TEMP)/X
            Z(I,J)=TEMP
            Z(J,I)=TEMP
25          CONTINUE
27        CONTINUE
30        CONTINUE
      RETURN
      END
      SUBROUTINE BOUNDS(Z,TRU,TRL,XV,NNS,NT)
      REAL Z(NNS,NNS),XV(NT)
      REAL*8 TRU(NT),TRL(NT),D,SUM
      DO 5 I=1,NT
        TRU(I)=0.D0
5       TRL(I)=0.D0
      DO 20 I=1,NNS
        D=Z(I,I)
        IF(D .LT. .0001) D=.0001
        SUM=(-1.)*D*D
        DO 10 J=1,NNS
10        SUM=SUM + Z(I,J)**2
        DO 15 J=1,NT
          TRU(J)=TRU(J) + (1. + SUM/D/XV(J))/(D + XV(J) + SUM/D)
15        TRL(J)=TRL(J) + 1./(D + XV(J))
20      CONTINUE
      RETURN
      END
      SUBROUTINE BOUNDA(Z,AINV,HOLD,LIST,WKB,TRU,TRL,XV,NNS,NT,NS2)
      REAL Z(NNS,NNS),AINV(NS2),HOLD(NNS),WKB(NNS),XV(NT)
      REAL*8 TRU(NT),TRL(NT),D,SUM
      INTEGER LIST(NNS)
C                             SUBROUTINE TO FIND APPROXIMATE TRACE OF
C                                AINV*(ZSZ + AINV*X)**-1
C                             NMAX IS LARGEST SEGMENT TO BE INVERTED
      DO 10 IT=1,NT
        TRU(IT)=0.D0
10      TRL(IT)=0.D0
      NMAX=(SQRT(1. + 8.*NNS) - 1.)/2
      DO 50 I=1,NNS
        D=Z(I,I)
        DO 27 J=1,NMAX
          LIST(J)=0
27        HOLD(J)=0.0
        NIS=0
        SMALL=0.0
        DO 30 J=1,NNS
          IF(I .GE. J) IND=I*(I-1)/2 + J
          IF(I .LT. J) IND=J*(J-1)/2 + I
          T=ABS(AINV(IND))
          IF(T .LE. SMALL) GO TO 30
          IF(NIS .EQ. NMAX) GO TO 28
```

```
          NIS=NIS + 1
          HOLD(NIS)=T
          LIST(NIS)=J
          IF(J .EQ. I) INDB=NIS
          GO TO 30
28        SMALL=99999.
          DO 29 K=1,NMAX
            IF(HOLD(K) .GE. SMALL) GO TO 29
            SMALL=HOLD(K)
            INDS=K
29          CONTINUE
          HOLD(INDS)=T
          LIST(INDS)=J
          IF(J .EQ. I) INDB=INDS
30        CONTINUE
        DO 45 IT=1,NT
          X=XV(IT)
          IND=0
        DO 33 J=1,NIS
          J2=LIST(J)
          DO 33 K=1,J
            K2=LIST(K)
            IF(J2 .GE. K2) IND2=J2*(J2-1)/2 + K2
            IF(J2 .LT. K2) IND2=K2*(K2-1)/2 + J2
            IND=IND + 1
33          HOLD(IND)=Z(J2,K2) + AINV(IND2)*X
        IJOB=1
        CALL LINV3P(HOLD,WKB,IJOB,NIS,IER)
        SUM=0.DO
        LB=LIST(INDB)
        DO 40 J=1,NIS
          IF(J .LE. INDB) IND=INDB*(INDB-1)/2 + J
          IF(J .GT. INDB) IND=J*(J-1)/2 + INDB
          LJ=LIST(J)
          IF(LJ .GE. LB) IND2=LJ*(LJ-1)/2 + LB
          IF(LJ .LT. LB) IND2=LB*(LB-1)/2 + LJ
40        SUM=SUM + AINV(IND2)*HOLD(IND)
        TRL(IT)=TRL(IT) + SUM
        TRU(IT)=TRU(IT) + 1./(D + X)
45      CONTINUE
50      CONTINUE
      RETURN
      END
      SUBROUTINE MULT(N1,N2,N3,A,B,C,IB,IC)
      REAL A(N1,N3),B(N1,N2),C(N2,N3)
      REAL*8 SUM
C                           MULTIPLICATION TO GET  A = B * C
C                           OR, IF IB=1            A = B'* C
C                           OR, IF IC=1            A = B * C'
      IF((N1-N2)*IB .NE. 0) GO TO 99
```

```
      IF((N2-N3)*IC .NE. 0) GO TO 99
      DO 20 I=1,N1
        DO 20 J=1,N3
          SUM=0.DO
          IF(IB+IC .NE. 0) GO TO 9
            DO 5 K=1,N2
5             SUM=SUM + B(I,K)*C(K,J)
            GO TO 20
9         IF(IB .EQ. 0) GO TO 14
            DO 10 K=1,N2
10            SUM=SUM + B(K,I)*C(K,J)
            GO TO 20
14          DO 15 K=1,N2
15            SUM=SUM + B(I,K)*C(J,K)
20        A(I,J)=SUM
      RETURN
99    PRINT 100
100   FORMAT('0ATTEMPT TO MULTIPLY NON-CONFORMING MATRICES')
      STOP
      END
 //GO.FT10F001 DD UNIT=DISK,DSN=P.I6280.PASS,DISP=(OLD,KEEP)
 //GO.FT11F001 DD *
    1    1    1.
    2    2    1.
    3    3    1.
    4    4    1.
    5    5    1.
    6    6    1.
    7    7    1.
    8    8    1.
    9    9    1.
   10   10    1.
 //GO.FT12F001 DD DUMMY
```

Output from ABSORBM for Example Data Set

NO. OF RECORDS,FILLED SUBCLASSES,HERDS,AND SIRES

107		45	10	10	
N	SIRE ID	GROUP	#PROG	#HERDS	EFF. NUM.
1	1	1	4	1	3.11
2	2	1	7	3	5.47
3	3	1	8	4	6.29
4	4	1	9	3	7.10
5	5	1	15	6	11.76
6	6	2	12	5	7.84
7	7	2	10	4	8.02
8	8	2	21	8	15.72
9	9	2	14	7	9.61
10	10	2	7	4	5.00

Output from REMLM for Example Data Set

```
WSSUB ESTIMATES OF ERROR VARIANCES
    1       2.998805
    2       7.564082
 RATIOS FOR DIAGONAL     9.000014     9.000005
ROUND   1 SUCCESSFULLY COMPLETED.
 RATIOS FOR DIAGONAL    12.822980     4.954104
ROUND   2 SUCCESSFULLY COMPLETED.
 RATIOS FOR DIAGONAL    15.717701     3.690523
ROUND   3 SUCCESSFULLY COMPLETED.
 RATIOS FOR DIAGONAL    17.822479     3.283190
ROUND   4 SUCCESSFULLY COMPLETED.
 RATIOS FOR DIAGONAL    19.325287     3.148664
ROUND   5 SUCCESSFULLY COMPLETED.
 RATIOS FOR DIAGONAL    20.398026     3.103594
ROUND   6 SUCCESSFULLY COMPLETED.
 RATIOS FOR DIAGONAL    21.170120     3.088619
ROUND   7 SUCCESSFULLY COMPLETED.
 RATIOS FOR DIAGONAL    21.731857     3.084010
ROUND   8 SUCCESSFULLY COMPLETED.
 RATIOS FOR DIAGONAL    22.144745     3.083013
ROUND   9 SUCCESSFULLY COMPLETED.
 RATIOS FOR DIAGONAL    22.450882     3.083244
ROUND  10 SUCCESSFULLY COMPLETED.
 RATIOS FOR DIAGONAL    22.679474     3.083840
ROUND  11 SUCCESSFULLY COMPLETED.
 RATIOS FOR DIAGONAL    22.851135     3.084493
ROUND  12 SUCCESSFULLY COMPLETED.
 RATIOS FOR DIAGONAL    22.980530     3.085078
ROUND  13 SUCCESSFULLY COMPLETED.
 RATIOS FOR DIAGONAL    23.078369     3.085567
ROUND  14 SUCCESSFULLY COMPLETED.
 RATIOS FOR DIAGONAL    23.152481     3.085970
ROUND  15 SUCCESSFULLY COMPLETED.
 RATIOS FOR DIAGONAL    23.208755     3.086289
ROUND  16 SUCCESSFULLY COMPLETED.
 RATIOS FOR DIAGONAL    23.251495     3.086542
ROUND  17 SUCCESSFULLY COMPLETED.

 FINAL ESTIMATES OF ERR VAR.,SIRE VAR, AND HERITABILITY
    1       3.624846      0.998634      0.863967
    2       8.643206      0.914865      0.382866

 STANDARD ERRORS OF CORRESPONDING PARAMETERS ABOVE
    1       0.651041      0.793494      0.551793
```

2 1.552365 1.091537 0.417733

TRAITS		PHENOTYPIC COV. AND CORR.		ERROR COV. AND CORR.	
2	1	-1.05342	-0.15847	-0.36096	-0.06449

	GROUP	EFF.NUM.	GROUP SOLUTIONS	
1	1	20.05	-0.544298	-1.606435
2	2	20.05	0.0	0.0

STANDARD ERRORS OF GROUP SOLUTIONS

1	1	20.05	0.470884	0.699629
2	2	20.05	0.0	0.0

NOTICE: ETA'S CONTAIN GENETIC GROUP SOLNS

SIRE ID	GRP	EFF.NUM.	ESTIMATED TRANSMITTING ABILITIES	
1	1	2.95	-0.602522	-1.442743
2	1	4.64	0.099539	-2.402162
3	1	5.59	-0.635067	-1.103045
4	1	6.61	0.174076	-2.248347
5	1	9.09	-1.757678	-0.835747
6	2	7.59	-0.969330	0.706570
7	2	7.05	0.073009	-0.061414
8	2	11.82	-0.592271	0.037470
9	2	9.15	0.137395	0.522709
10	2	4.88	1.353229	-1.207040